Taking distributed representations seriously in functional brain imaging

Christopher R. Cox and Timothy T. Rogers

University of Wisconsin, Madison

[crcox@wisc.edu](mailto:crcox@wisc.edu) (Corresponding author)

ttrogers@wisc.edu

W J Brogden Psychology Building

1202 West Johnson Street, rm 238

Madison, WI 53706

Author’s Note

We would like to acknowledge Nikhil Rao and Robert Nowak, for developing the SOS Lasso loss function and its MATLAB implementation, and Mark Seidenberg for valuable discussions of this work in its development.

We have no conflicts of interest.

Keywords: distributed representations, PDP, MVPA, SOS Lasso, fMRI

ABSTRACT

The Parallel Distributed Processing (PDP) approach to cognition assumes that active mental representations are distributed patterns of neural activation, with each neural population contributing to many different representations and each representation involving many populations. Distributed representations of this kind provide the basis for influential models in many different domains of cognition. Until recently, however, the PDP view of representation has not strongly influenced functional brain imaging, which has, for historical and methodological reasons, tended to adopt modular assumptions about how the brain encodes information. New methods for multivoxel pattern analysis (MVPA) relax these but rely on other important assumptions about the nature of the underlying representations that sometimes go unspecified. In this paper we consider how analysis of functional brain imaging data is best approached if the PDP assumptions about representation are valid. To this end, we compare and contrast five different data analysis methods in their ability to discover distributed representations in artificial data generated from a simple neural network model. We illustrate how the assumptions underlying each method lead to bias in the nature of signal detected, and show that some methods will fare better at discovering the relevant information if the PDP assumptions are valid. We then compare performance of the different methods in analysis of one well-known publicly available fMRI dataset, with results suggesting that, at least in this case, the core representations do indeed have the properties assumed by PDP. The results suggest a general strategy for fMRI data analysis that may provide a fuller picture of how information is represented and coded in neural activity, and may allow for a better connection between mechanistic models of cognitive function and data arising from functional brain imaging.

Taking distributed representations seriously in functional brain imaging

An important question for cognitive science concerns the nature of mental representations. What are they, how are they structured, where do they come from, and how do cognitive processes operate on them to support behavior? One approach to cognition, variously referred to as the parallel distributed processing (PDP), connectionist, or neural network approach, has offered fairly specific answers to these questions (McClelland & Rumelhart, 1986). It begins with the central tenet of cognitive neuroscience that all cognitive processes ultimately arise from the propagation of activity amongst large populations of neurons communicating their states via excitatory and inhibitory synaptic connections; but it further proposes that the important characteristics of such systems can be captured in simplified abstract form by computer models that simulate the flow of information through networks of neuron-like processing units connected by weighted synapse-like connections. All cognitive abilities are proposed to arise from representational and processing mechanisms that can be so described and understood. Accordingly, the framework offers a constrained view of what mental representations are: patterns of neural activity evoked by a stimulus or a process over a set of units operating in parallel. Often such representations are proposed to be highly distributed, so that (1) any given unit contributes to the representation of many different items, (2) any given item representation is encoded over many different units and (3) the representation inheres in the full pattern of activation over all units, and not in the activation of individual units.

Distributed representations of this kind have several properties that make them useful for understanding various aspects of cognition. For instance, they provide a natural basis for similarity-based generalization (Hinton, 1984; Rumelhart & Todd, 1993). Two different items that generate overlapping patterns of activation over the same units will tend to produce similar responses downstream in the network. Thus distributed representations explain how prior learning supports the processing of novel inputs, an ability central to accounts of categorization, inductive inference, language processing, and many other cognitive phenomena. Similarity-based generalization also naturally produces patterns of behavior observed in many different tasks, such as typicality effects, frequency effects, and effects of quasi-regularity (Plaut, McClelland, Seidenberg, & Patterson, 1996; Rogers & McClelland, 2004). Distributed representations explain why, with neuropathology, cognitive abilities are not disrupted in an all-or-none fashion, but instead degrade gracefully: when some units in the representation are destroyed or disrupted, the remaining units continue to communicate with downstream units, transmitting information that may still be at least approximately correct (Allport, 1985; Cooper & Shallice, 2011). They provide a means of understanding the acquisition of new representations: rather than adding new information to a database, or new representational elements into a processing system, new representations can be acquired by adjusting connections within the network so that a given stimulus or process generates a new pattern of activity over the existing units (Rumelhart & Todd, 1993). Finally, distributed representations can be highly efficient. With a local code, in which each unit represents one and only one item of information, then with n units it is possible to represent just n items. With a binary distributed coded, the same n units can represent distinct items; and if the units encode continuous activation values there is, in principle, no limit to the number of items a given set of units can encode (Hinton, 1984).

These virtues are well known and form the basis for influential accounts of healthy, disordered, and developing cognition in theories of reading (Seidenberg & McClelland, 1989; Harm & Seidenberg, 2004), inflectional morphology (Rumelhart & McClelland, 1986; Joanisse & Seidenberg, 1999; Plunkett & Juola, 1999), semantic processing (Farah & McClelland, 1991; Rogers & McClelland, 2004; Rogers et al., 2004), routine sequential action (Botvinick & Plaut, 2004) and many other domains (see Rogers & McClelland, 2014 for a recent review). Perhaps surprisingly, however, remarkably little work in cognitive neuroscience has attempted to directly test PDP's assumptions about the neural basis of mental representations. The reasons for this gap are probably methodological, at least in part. The most ubiquitous methods in human cognitive neuroscience---fMRI and other functional brain imaging technologies---typically yield vast amounts of noisy data. To discern interesting patterns in these datasets, or to test particular hypotheses, the statistical models employed must adopt specific assumptions about the nature of the underlying signal. For many years, standard statistical approaches were built upon representational assumptions that were at odds with those adopted under the PDP approach (Kriegeskorte, Mur, & Bandettini, 2008). As a result it has been difficult to relate the results of such analyses to the predictions of PDP models. This has begun to change with the advent of new multivariate methods for analyzing brain imaging data but many challenges still remain. Several different approaches have appeared in the literature in recent years (Pereira, Mitchell, & Botvinick, 2009; Kriegeskorte, Goebel, & Bandettini, 2006; Kriegeskorte & Kievit, 2013; Mitchell et al., 2008), each carrying with it particular assumptions about the nature of the underlying neural code, and so bringing biases in the kind of signal it is capable of detecting. Moreover, the nature of the assumptions and their implications for measuring neural representations may not always be completely transparent, although important work is being done to address this (e.g., Davis & Poldrack, 2013). Thus, although the PDP framework and related approaches put forward quite specific hypotheses about the nature of representations in the mind and brain, it has not been clear how neuroscientific methods, and in particular functional brain imaging, might best be leveraged to test these hypotheses.

The goal of the current paper is to consider how the analysis of brain imaging data might best be approached if the PDP assumptions about mental representation are valid. We first consider in more detail what the PDP assumptions are and why they pose challenges for standard and even many state of the art brain imaging methods. To illustrate these points, we compare and contrast the results yielded by five different statistical methods in the analysis of the activation patterns generated by different inputs to a simple PDP model. Since the behavior, architecture, and representational structure embedded in the simple model are fully known, it is possible to measure the extent to which the various methods succeed in identifying the model components that encode interesting representational structure. This analysis thus illustrates the strengths and weaknesses of different approaches if PDP assumptions about representation are valid. The results suggest a new strategy for the analysis of functional imaging data that may help to better connect PDP models to cognitive neuroscience. We then assess the utility of this approach by comparing its results to those of other state of the art methods in the analysis of one well-known publicly released fMRI dataset.

# A brief overview of PDP models and their representational assumptions

PDP models are composed of simple processing units that communicate via weighted synapse-like connections (McClelland & Rumelhart, 1986; Rogers & McClelland, 2014). Each unit adopts an activation state, typically varying between 0 and 1, that can be viewed as analogous in some respects to the mean firing rate of a population of spiking neurons proportional to their maximal rate (Zipser & Andersen, 1988). Units transmit information about their current activation through weighted connections, which can be viewed as capturing the net effect of activity in one population of neurons on another. Weights are typically real-valued, with negative numbers indicating a net inhibitory effect and positive numbers indicating a net excitatory effect. Each unit computes a simple process: it adjusts its current activation state according to the input it receives from other units in the networks. If a given receiving unit receives inputs from a set of sending units, then the input is usually computed as the inner product of the activation across all sending units and the values of the weights projecting from the sending units to the receiving units. The unit then converts the net input into a new activation state according to a specified transfer function (often a sigmoid function of the net input). All units are conceived as computing inputs and updating activation states in parallel in continuous real time (hence “parallel” distributed processing), though on serial computers this parallel process is simulated by updating units in discrete steps in randomly permuted order.

Within a network, units are generally organized into layers, which govern the overall connectivity of the network: units within a layer tend to receive connections from, and direct connections toward, a similar set of units elsewhere in the network. Typically a subset of the units are specified to receive inputs directly from sensory systems (or other input systems outside the model), and to direct outputs toward motor systems (or other output systems outside the model). These unit subsets encode the input provided to the model and the outputs that simulate the model response. They are often referred to as *visible* units, because the theorist directly stipulates how different stimulus events and behaviors are represented with patterns of activation over the input and output units. Most models also include sets of units whose inputs and outputs are directed only to other units contained within the model---they do not receive external inputs from or direct outputs toward the model environment. For these *hidden* units, the theorist does not stipulate how different stimulus events or behaviors are to be coded with patterns of activation. Instead, the patterns of activation that arise across these units are determined solely by the values of the interconnecting weights.

The weights themselves are viewed as being shaped by learning and experience. Many different learning algorithms have been explored in this framework (see Hinton, 2014), but all share the general idea that the weights gradually change over time in order to optimize some objective function---for instance, minimizing the discrepancy between the outputs the model generates and the correct “target” outputs---as the network processes information from different stimulus events. Because the weights adapt to experience, and because the patterns of activation over hidden units depend upon the weight values, PDP models are therefore capable of acquiring learned internal representations: the patterns of activation generated over hidden units by a given stimulus after the network has undergone learning in a model environment. One interesting aspect of PDP models, responsible for their utility in many different cognitive domains, concerns the nature of the internal representations they acquire after learning in a structured environment. Often the models can acquire internal representations that may seem counter-intuitive from other points of view, but that can be shown, through computer simulations, to support behaviors documented in the domain of interest. Figure 1 and its caption provide one example of a PDP model used to understand aspects of semantic memory.

**---Figure 1 about here---**

With this overview of how PDP models work, we are ready to consider the challenges that the framework raises for the discovery of mental representations in functional brain imaging data. Many difficult issues arise, of course, in any effort to relate artificial neural networks to real neural networks. Because network models are functional abstractions of the neural processes they aim to uncover, they necessarily gloss the complexity, and many aspects of structure and behavior, known to be important in real nervous systems. Whereas individual neurons exhibit all-or-nothing spiking behavior, model units assume continuous activation states. Low-level dynamics such as lateral inhibition, temporal coherence, and local extra-cellular conditions are glossed over in most connectionist models, while morphological differences among neuron types, cytoarchitecture, and other facts about brains are completely abstracted away. PDP units can instead be viewed as capturing, in a modest number of processing elements, the same informational states existing across vast numbers of heterogeneous spiking neurons in real nervous systems (Smolensky, 1986; Rogers & McClelland, 2014). The central assumption is that the representational content and cognitive functions expressed in the coordinated spiking behaviors of hundreds or thousands of neurons can be usefully approximated as a much smaller vector of continuous-valued activations. As we have noted elsewhere (Cox, Seidenberg, & Rogers, 2014), this is essentially the same assumption adopted in fMRI and other brain imaging methods which summarize the dynamical activity over hundreds of thousands of neurons at a millisecond timescale with a much smaller vector of real-valued numbers, each expressing the overall metabolic demands exerted by populations of neurons within a voxel. The effort to relate neural activity to cognitive events entails the assumption that important informational states over vast sets of neurons can be so abstracted.

We therefore adopt here a fairly simplified stance on the relationship between network models and the brain networks we seek to discover in imaging data. Specifically, we assume that the activation of a single unit in a network model is roughly analogous to mean neural activity in a population of hundreds of neighboring neurons within a small volume of cortex as estimated at a single voxel from BOLD activity in fMRI. Thus we will treat the pattern of activation generated by a given stimulus over units in a model network as analogous to the set of beta coefficients estimated over voxels from the BOLD response evoked by a given stimulus in a sparse event-related design.

Even with this relatively transparent view of the relation between model elements and measured physiological responses, PDP raises four difficult challenges for the discovery of representational structure in the brain.

1. *The behavior of a given cortical subregion (i.e., voxel or voxel cluster) can vary substantially across individuals even if different individuals encode the same representational structure across the same general regions.* For any given network, there are typically many different weight configurations that can generate appropriate outputs given the various inputs. The particular configuration that a network discovers with learning can depend on many things, including the initial random weight configuration, the ordering and distribution of the learning experiences sampled from the environment, and the effects of noise in the unit activations and/or weight changes. Thus a particular hidden unit in a given model can, across different training runs in the exact same environment, exhibit quite different patterns of activation in response to a given input. Yet the internal representations learned by a network are not arbitrary; the learning models are of interest because they reliably extract important similarity structure across the set of input and output patterns to which they are exposed. What varies is the particular way that individual units contribute to encoding the interesting structure across network runs. Cox et al. (2014) provide a simple example of this kind of variability in a simple model.

We can conceive of a single model training run as simulating the effects of learning and experience in a single individual person. The different weight configurations and internal codes that arise across model training runs thus indicate the kind of variability in representation that may exist across individuals under the PDP view, even if the individuals show the same pattern of overt behavior in the domain and the same gross neural architecture. Specifically, the response generated by a given stimulus or process in a given patch of cortex may vary arbitrarily across individuals, even if the same representational structure is being encoded across the same cortical subregions. This possibility poses a challenge to imaging methods that focus on finding voxel clusters that reside in similar locations and respond in similar ways across individuals. If representations vary across individuals in the way that PDP models suggest, such methods will fail to discover them.

This consequence of distributed representations may pose greater problems for finding signal in some cortical regions than others. In peripheral regions (i.e., early sensory and motor cortices), it is clear that information is encoded in largely the same way, and with a largely similar neuroanatomical organization, across individuals. In association cortices, it may be that neural codes are less constrained are more strongly shaped by learning and experience, so that the way information is organized across cortex is more highly variable. PDP models provide a rough analog to this state of affairs, insofar as input and output units for a given model are stipulated to represent information in the same way in every model training run---that is, in every model “individual.” The issues of variability in representation mainly apply to learned internal representations coded across hidden units.

1. *Activation of individual units may not be interpretable independent of other units.* A corollary of the preceding points is that the behavior of a given cortical unit may not be interpretable, or may have quite different interpretations across individuals, when analyzed independently from other units. This property of distributed representation is important because it suggests that univariate approaches to data analysis---methods that assess the behavior of individual voxels or voxel clusters independently---can fail to uncover important components of neural representations. Wherever the interesting structure is embedded in activations across multiple cortical units, but is not reflected in individual units, such methods will yield null results (see Cox et al. 2014 for a concrete example).
2. *The functional model architecture may not map transparently onto anatomical structure in the brain.* A third issue concerns the relationship between the functional architecture of a computational model used to simulate performance on a task of interest and the actual anatomical structure of the corresponding neural network in the brain. As noted earlier, units in PDP models are organized into layers, with units in a given layer receiving connections from and directing connections toward the same subsets of units elsewhere in a network. The layer is a useful construct for understanding how a network functions, insofar as the units within a layer, by virtue of having similar connectivity to the broader network, “work together” to represent and process the same information. Distributed internal representations in PDP networks are typically viewed, therefore, as being encoded across units within a particular layer.

It may seem natural to view layers as model analogs of cortical regions, so that the gross architecture of a computational model maps transparently onto the anatomical structure of networks in the brain that carry out the modeled cognitive function. Though this analogy is reasonable, it is not the only possible way that the functional architecture of a computational model might relate to the neuroanatomical structure of a corresponding cortical network. In fact, the layers of a computational model do not, in principle, have any implications for how the corresponding cortical units might be anatomically situated in the brain. Units that function together as a “layer” could be situated in multiple different cortical regions, or widely dispersed anatomically, or interdigitated with other units subserving different functions. The defining property of layers in a computational model is their pattern of connectivity in the gross architecture, and the same network connectivity can exist among many different spatial arrangements of units. In other words, the relationship between the functional architecture of a computational model---the grouping of units into layers as typically depicted in model figures, for instance---may not transparently reflect the topological arrangement of the corresponding cortical units in the brain.

This lack of transparency poses a problem for approaches to brain imaging that assume representations to be encoded over a volume of anatomically contiguous cortical units, including approaches that average signal over regions of interest, that spatially blur signal, or that restrict statistical analysis only to voxels within pre-specified areas. If cortical units that function together as a representational substrate do not happen to reside in a single contiguous cortical region, such methods may fail to discover important signal.

This is not to suggest that the PDP view predicts that anatomical structure is unimportant, or that shared structure across individuals is unexpected or meaningless. To the contrary, the connectivity of a given network strongly constrains its behavior. Thus the network architecture always constitutes an important aspect of the explanatory hypothesis a model is intended to exemplify. It is typically assumed that this architecture is largely shared across individuals, and that, however it is anatomically situated in the brain, there will be at least coarse similarities across individuals.

1. *The network of interest in any given study co-exists in the brain with many other networks, all subserving other functions that may not be of interest.* Any given computational model is designed to aid understanding of a particular aspect of cognition, and typically includes only those elements that the theory stipulates to be important for the behavior of interest. Even if the model is a relatively faithful and accurate abstraction of a real cortical network in the brain, the physiological measurements generated by that network will be intermingled with measurements from a great many other cortical systems involved in other aspects of cognition unrelated to the task of interest. Odds are that the great majority of measurements taken will reflect metabolic activity unrelated to the representational structure we are searching for. Thus the effort to find distributed representations in brain imaging data raises concerns about needles and haystacks.

## Summary

The representational assumptions of the PDP framework lead to rather bleak outlook. The behaviors of individual cortical units (i.e., voxels) may not independently covary with or otherwise reflect the objects of representation we are interested in finding in a systematic way across individuals. Mental representations may instead inhere in the patterns of activation evoked across whole sets of units that together function as a representational ensemble by virtue of their connectivity within the overall cortical network (like the layers of a neural network model). This is the core sense in which representations are distributed in the PDP view. The way that a particular unit contributes to different representations can be highly variable across individuals, even if the ensemble encodes the same representational structure across individuals. This means that the search for voxels exhibiting similar responses in similar anatomical locations across people will fail to reveal important representational structure. Moreover, the units that operate together as a representational ensemble may not be anatomical neighbors, may vary in their location to some extent across individuals, and are certain to be buried within the mountain of measurements provided by functional imaging technologies across the whole brain. These possibilities raise a daunting challenge: representational structure can only be discerned by considering the whole pattern of activation over a representational ensemble; yet the units within such an ensemble may be anatomically dispersed and intermingled with a vast amount of irrelevant information. One cannot understand the representation without knowing which units together constitute an ensemble, but how is one to find the ensemble in the first place?

In what follows, we assess how well different approaches to fMRI data analysis meet this challenge by applying them to the discovery of representational structure in data generated by a simple neural network model as it processes different input patterns. We will see that all methods bring with them important biases in the kinds of representational structure they are capable of detecting, and that some methods are better-suited to finding the distributed structure that the PDP framework assumes. We will also see that patterns of results across methods can provide important information about the nature of the representations encoded in different parts of a network.

# Model and Methods

**---Figure 2 about here---**

The model we will employ for these analyses is illustrated in Figure 2. It is an auto-encoder network: when presented with an experience in the form of a pattern of activity over its 36 input units, it learns to reproduce that same pattern over its 36 output units. Auto-encoder networks have been used as simple models of human memory, because once they have learned they are capable of both retrieving full information from a partial cue and of generalizing prior learning to new items (McClelland & Rumelhart, 1985). In this case, however, we do not intend the model to embody a specific hypothesis about a particular real-world cognitive function. Instead, it is designed to make explicit the challenges noted in the introduction.

To this end, the patterns that the model processes are viewed as coming from two different domains, A and B, corresponding to some cognitive distinction of theoretical import. For instance, A and B might correspond to nouns versus verbs, or animals versus manmade objects, or faces versus non-faces, or any other binary distinction thought to be of potential relevance to behavior. Each individual item is represented with a unique pattern of activation over input units, and the network's task is simply to generate the same pattern over output units. In this sense, there is no explicit representation of the two classes A and B in the inputs, outputs, or network task. The two domains are assumed, however, to be distinguishable from the distribution of input/output properties they possess. Specifically, one subset of input/output properties is marginally more likely to be active for items from domain A, while another subset is marginally more likely to be active for items in domain B. We will refer to these subsets together as *systematic I/O* units, because they each weakly co-vary with the representational distinction of interest. Each item also possesses several features coded by *arbitrary I/O* units whose activations do not systematically differ between domains.

After the model has learned, it is possible to “query” it by presenting an input pattern and generating patterns of activation throughout the rest of the network. As noted earlier, we take the activation at each unit in response to an input as a model analog of the neural response to a stimulus estimated from the BOLD signal at a single voxel in a single individual. Across different training runs, the model will always exhibit the same overt behavior (generating the correct pattern over output units), but arising from different configurations of weights, and hence from different internal representations. Variability in weight configurations and internal representations acquired across different training runs thus provides a model analog of individual variability in the neural representations acquired across the population. To simulate data generated by a functional brain imaging study with, say, 10 participants, we train the model 10 times with different random initial weight configurations. For each trained model, we record the pattern of activation generated over all model units by each input pattern (i.e., stimulus), taking these as model fMRI data. The question we then wish to ask, by applying different statistical methods to the analysis of this synthetic imaging data from a sample of trained models, is the following: which units in the network encode representations of the domains A and B, and how?

The network architecture is designed so that there are two possible answers to this question. The first answer is that representations of A and B are directly encoded in the individual activations of the systematic I/O units. For all input and output units, the response of a given unit to a particular item is directly specified by the environment, so that these units will always respond to a given stimulus in the same way across model individuals. Each systematic I/O unit has a marginally different probability of being active depending upon the domain; in this sense the A units each independently encode a representation of the A domain and the B units encode a representation of the B domain. The relationship between domain and activation is, however, stipulated to be quite loose: for each domain, only a small number of the corresponding systematic units will be active for any given item---each unit participates in just a few patterns. Each item thus overlaps in their systematic properties with just a few other items in the domain, and the correlation between activation and domain is weak for any individual unit. We further stipulate that the A input and output units are anatomical neighbors, as are the B input and output units, and that this anatomical arrangement is exactly the same across individuals. Thus the systematic I/O units individually encode a *weak* distinction between A and B that is *consistent* across model individuals and is *anatomically localized* within input and output layers.

The second answer is that the representations of A and B domains are encoded in a distributed fashion over a subset of model hidden units. As shown in the Figure, the input units project to the output units by way of two separate hidden layers. The *systematic hidden layer* (SH) contains 7 hidden units that receive connections from the systematic input units and send connections to the systematic output units. The *arbitrary hidden layer* (AH) also contains 7 units that receive connections from the arbitrary inputs, and send connections to *both* the systematic and arbitrary outputs. The weights are shaped by learning, so every input generates a pattern of activation---a learned internal representation---over both the SH and AH layers. The particular way that layers are connected, however, ensures that these internal representations will have specific representational properties. The SH layer connects systematic inputs to systematic inputs. Because items within a domain have a weak tendency to share systematic properties, the SH units can efficiently perform their mapping by representing the domain structure: items within a domain evoke similar patterns over units and items from different domains evoke quite different patterns. The AH layer receives inputs only from the arbitrary input units and directs outputs to all units. There is no tendency for items within a domain to share arbitrary features, so there is little pressure for these units to represent the domain structure. The AH layer thus acquires distributed internal representations that have little obvious structure. The weights in the arbitrary pathways effectively serve to “memorize” both the arbitrary features and the idiosyncratic differences among items in the same domain. In other words, the architecture produces a division of labor in which the SH layer learns distributed representations of the domain structure and the AH layer learns idiosyncratic differences among items. A good method, then, should identify SH units as important for representing the domains.

Indeed, the SH units arguably provide a better encoding of the domain structure than do the systematic I/O units. To illustrate this we first trained the model to saturation on 15 runs with different initial random weights, then analyzed, for each layer in the model, the Euclidean distance between the patterns of activation elicited by each pair of stimuli in the model. For each layer, we computed the mean distances for pairs within a domain and for pairs in different domains. We then took the ratio of between-domain to within-domain distances as a measure of how well the domain structure is expressed in each layer. A ratio of 1 indicates that between- and within-distances are about the same; a number greater than 1 indicates that between-domain distances are larger on average than within-domain differences, indicating good differentiation of the domains. The results averaged across the 10 model subjects are shown in Figure 3. For arbitrary units (both I/O and hidden), no domain structure is expressed: both ratios are near 1. For systematic I/O units, the ratio is clearly larger than 1, indicating reasonable encoding of the domain structure, but the ratio is much larger for the SH units, indicating that these distributed representations do a better job of systematically differentiating items from the two domains.

Finally, the model also includes 22 completely irrelevant units. These are units assumed to be anatomically near the SH and AH units but uninvolved in the task. These units always take a low activation value, and provide a simple model analog of the fact noted above that the units of interest may exist alongside other units that remain uninvolved in the task under investigation.

**---Figure 3 about here---**

With this general understanding of the model behavior, let's consider how it makes explicit the four challenges for brain imaging noted earlier. First, although the SH units jointly encode the same representational structure across model subjects, the contribution of a given hidden unit to this structure varies arbitrarily across model subjects (challenge 1). Second, the mean activations of SH units do not systematically differ for items in the A and B domains: to find the important structure, one must consider the pattern evoked over multiple units (challenge 2). Third, the functional architecture of the model shown in Figure 2 can be anatomically arranged in many different ways (challenge 3). To make this issue explicit, we consider two different topographic arrangements of the functional model. In the first, units within the same layer are always situated as anatomical neighbors, so that the representations encoded by the SH and the AH layers are *anatomically localized*. In the second arrangement, we assume that the SH units are *spatially intermingled* with the AH units, in a different way across model individuals, so that the representations they encode are *anatomically dispersed*. In the results we will consider how well each method identifies the SH units as a function of whether they are localized or dispersed. Finally, the model captures the idea that the units of interest constitute only a small proportion of all the units measured (challenge 4). In the model itself, most of the units encode information irrelevant to the stimulus domain (the 43 arbitrary units plus 22 irrelevant units). The next largest set are the 36 systematic I/O units that encode the domain structure weakly but consistently across subjects. The units of greatest interest, the 7 SH units, constitute just 6% of all the units in the data.

## Summary

Though very simple, this auto-encoder network captures each of the challenges noted in the introduction: it acquires distributed internal representations that express representational structure of interest; the way the structure is coded across units varies in different individual models; the structure cannot be discerned from the activations of single units but arises in patterns over multiple units; the relationship between the functional architecture and the underlying model topography can be opaque; and the units that encode the structure we wish to discover are buried in a large number of other measurements. The question we now address is how well different analysis methods fare at discovering representational structure across both systematic I/O units and the SH units, when they are applied to data generated from a sample of model training runs.

# Simulation details

The model shown in Figure 2 was trained on 72 items sampled from two domains, A and B. Each item activated exactly 2 systematic and 2 arbitrary input units, and across items each unit was active in exactly 8 items. Half of the systematic units were activated only by items from domain A, while the remaining half were activated only by items from domain B. Thus any pair of items in the same domain had a small probability of overlapping in some of their systematic properties, while items from different domains never overlapped in their systematic properties. Arbitrary units were equally likely to be active for items from domain A versus B.

The model was fit in LENS (Rohde, 1999) using back-propagation to minimize cross-entropy error. The weights were adjusted with a learning rate of 0.1, using momentum (“Doug's” momentum = 0.9) and subject to weight decay (decay constant = 0.001). The model was trained 10 times to asymptotic performance with very low error over 1000 epochs. Prior to each training run, the model was initialized with random weights sampled from a uniform distribution in the range [-1,1]. These 10 models were used to generate data for 10 model “subjects,” based on the patterns of activity elicited by each input over the whole network. Each model was presented with the 72 input patterns in sequence, and the pattern of activation elicited over the 98 units in the network (including the 22 irrelevant units, which always had an activation of zero) was recorded. The dataset for each model subject thus consisted of a matrix with 72 rows corresponding to stimulus items and 98 columns corresponding to model voxels. Each matrix contained the “true” response pattern for each subject to each item. To simulate noise in the measurement of this activity, a random value sampled independently from a Gaussian distribution with a mean of zero and standard deviation of 1 was added to each cell of the matrix. We take the resulting values in each cell of a matrix to be a model analog of the estimated BOLD response to a single stimulus at a single voxel in a single subject in an fMRI study.

To apply different brain-imaging methods to the discovery of structure, it is necessary to further stipulate the anatomical locations of the different units the model. In all simulations, input units were situated all together, with domain-A units neighboring one another, domain-B units neighboring one another, and arbitrary units neighboring one another. Output units were organized the same way, though outputs were assumed to be anatomically distal to inputs. The anatomical arrangement of input and output units was assumed to be identical across model individuals. For hidden units, we considered two different anatomical organizations. For *anatomically localized* models, units within a layer (SH, AH, or irrelevant) were also assumed to be anatomical neighbors, localized in the same way across model individuals. In the *anatomically dispersed* condition, units from the three hidden layers were assumed to be randomly intermingled with one another anatomically, in a different manner across model individuals. In either case, units in the hidden layers (together with irrelevant units) were assumed to be anatomically distal from both the input and output layers. For each anatomical variant the activation patterns evoked across model units by different inputs, and the ways these patterns were distorted by measurement noise, were identical---all that differed was the assumption about the spatial locations of the units in each layer.

# Results

With this understanding of the model, we are now ready to consider how different statistical methods for fMRI fare at discovering the model units that encode representations of the two domains, both in the case where the hidden units are anatomically localized and when they are anatomically dispersed. The methods we consider include the standard univariate contrast method and four forms of multivariate pattern classification (MVPC). Each method faces the challenges inherent in fMRI analysis---that of finding meaningful signal within a vast amount of quite noisy data. To address the challenge, each method adopts a different set of assumptions about the nature of the underlying signal, and so brings with it biases in the kinds of results it yields. For each method, we will begin with a brief exposition of the basic logic and essential concepts and will explicitly note the underlying representational assumptions. We then report the implementational details and results of the analysis, with the aim of answering four questions:

1. Does the method identify the systematic I/O units, but not arbitrary units, as important for domain representation?
2. Does the method identify the systematic hidden units, but not arbitrary units, as important for the domain representation?
3. Do the results differ when hidden units are anatomically localized versus dispersed?
4. Does the method indicate differences in how the information of interest is coded across unit sets? Specifically, does it indicate that some units respond more to A items than to B items, others show the reverse pattern, and still others express the A/B distinction with a distributed code?

## Univariate contrast analysis

Concept and assumptions. The univariate contrast analysis is the standard method for interrogating fMRI data. Its goal is to identify regions of cortex that, across subjects, exhibit systematically different mean BOLD responses to two (or more) different kinds of cognitive events. Typically the BOLD signal is spatially smoothed, so that the raw response at each voxel is replaced with a weighted average of the responses from anatomically neighboring voxels. The smoothed time-series is then modeled independently at each voxel for each subject using a deconvolution procedure. This yields a beta coefficient for each experiment condition at each voxel indicating how well the measured BOLD signal matches the response expected if the activation of neurons within the voxel varies systematically with the experiment condition. The beta coefficients for each subject are projected into a common anatomical reference space, and univariate statistical tests are computed at each voxel independently to assess whether the coefficients differ reliably in the two experimental conditions across subjects. Voxels that show significantly different responses across subjects are viewed as important for coding the representation of interest.

A major challenge for the approach lies in establishing a meaningful criterion of significance in the context of tens or even hundreds of thousands of individual statistical tests. To avoid both false-positives and punishing corrections for multiple comparisons, it is common to seek ways of reducing the number of tests performed. Several different methods have been employed, but all rely on the idea that the representations of interest can be localized to particular cortical regions, and that the responses of voxels within a functional region will be largely similar. With these assumptions, the number of tests can be reduced by (1) conducting regions-of-interest analyses, where the responses of voxels within a ROI are averaged and the test is performed on the result mean response, (2) applying cluster-thresholding, where tests are only performed on clusters of *n* anatomically contiguous voxels all showing a similar response across subjects, or (3) applying a topographic control of the false-discovery rate.

The univariate contrast method thus favors the discovery of clusters of anatomically neighboring voxels located in similar regions across individuals and showing similar response profiles across experimental conditions.

From this brief description we can see that the method relies on five assumptions about the nature of the neuro-cognitive representations, summarized in the “univariate” column of Table 1.

**---Table 1 about here---**

Implementation. The activity at each unit was modeled simultaneously for all subjects in a mixed effects model that treated subject as a random factor (Chen, Saad, Britton, Pine, & Cox, 2013; Friston, Stephan, Lund, Morcom, & Kiebel, 2005) using the lme4 package in R (Bates, 2013). Each model contains a single regressor, coding whether each item is an example of category A or B. The coefficients obtained from the mixed effects model were tested for significance using the Kenward-Roger approximation for the degrees of freedom (Kenward & Roger, 1997) and a standard F-test, numerator degrees of freedom = 1, denominator degrees of freedom = 9. The results are directly analogous to a repeated-measures ANOVA. The criterion for significance, alpha, is corrected to control the false discovery rate at q<0.05. The analysis was conducted for both the anatomically localized and the dispersed model. In both cases, the data were spatially smoothed, taking a weighted average over a three unit window, where the center unit was weighted about twice as much as the two flanking units.

**---Figure 4 about here---**

Results. Figure 4 shows the results of applying the univariate method to the localized (left) and dispersed (right) models. In these plots, each bar corresponds to a single unit in the model. The bars are ordered according to their functional role in the network, as indicated by the X-axis labels. Colored bars indicate units showing statistically significant differences in mean activation across model individuals, while grey bars indicate units that did not show significant differences. Among the colored bars, red indicates units where activation was systematically higher for domain A across models, and blue indicates units where activation was systematically higher for domain B. Note that, in the anatomically dispersed plot (right), the units are shown in their standard functional location for ease of interpretation.

In both localized and dispersed cases, the univariate contrast method identifies systematic I/O units as important for representing the A/B distinction, and correctly indicates that different subsets of input units code this information differently (some responding more to A than B and others showing the opposite pattern). Note that these are the units for which the five univariate assumptions about representation listed in Table 1 are all valid. In both localized and dispersed cases, however, the analysis completely misses the systematic hidden units, even though these jointly encode a cleaner representation of the A/B domain structure. The failure arises because, in both cases, the univariate assumptions are invalid. When hidden units are localized, assumptions 2 and 4 are violated: the way individual units encode information can vary across SH units in the same model individual, and across individuals at the same anatomical location. When the units are anatomically dispersed, assumption 3 is also invalid: the representation is coded in different anatomical locations across individuals. Because of these departures from the statistical assumptions, the mean activation of a unit at a given anatomical location across individuals does not differ reliably for SH units, even though these do reliably encode the domain distinction in each individual. These results are summarized in the column labeled “univariate” in Table 2.

Multivariate pattern classification. The remaining methods we consider are all variants of multi-voxel pattern analysis (MVPA) that rely on pattern classification algorithms (Norman, Polyn, Detre, & Haxby, 2006). Such approaches reverse the objective underlying univariate analyses: rather than using knowledge of the experimental design to explain variance in neural activity at individual voxels, MVPA uses the variance of neural activity across many voxels to make predictions about the experimental condition to which each trial, stimulus, or time point (henceforth, “example”) belongs (Mitchell et al., 2004; Pereira et al., 2009). To accomplish this, a classification algorithm is applied to a set of *training data* which include (a) the pattern of estimated activation evoked over a set of voxels for each of many examples and (b) a set of *labels* indicating the experimental condition or class associated with each pattern. For instance, in our model experiment, items in condition A might be labeled with a 0 while items from domain B are labeled with a 1. From the training data, the algorithm returns a pattern classifier---a statistical model that can be used to predict the label associated with any pattern of activation over voxels. Many different classification algorithms exist in the literature; as just one example, a logistic classifier will return a set of *weights*, one for each voxel, such that the estimated voxel activation, multiplied by its weight, summed over all voxels, and subject to a transformation function, yields a number that indicates the pattern label. In our example, a good logistic classifier should yield a number near 1 for all condition A items and near zero for all condition B items.

Even if there is no real signal at all in the data, it may be possible for a classifier to generate correct predictions for all items in the training set, especially when there are many predictors. Training set performance thus does not indicate whether the classifier is exploiting real signal in the data. Instead, the classifier is typically assessed on a *hold-out set*: an additional set of examples and labels collected in the same experiment but excluded from the training data. The classifier learned from the training data is applied to patterns in the hold-out set, and for each pattern it generates a “guess” about the associated condition label. The classifier output is compared to the true label to get a measure of accuracy. If a model performs above chance at classifying the hold-out set, this indicates that it is likely exploiting real information in the data. To ensure that the results do not depend upon the particular items chosen for the training and hold-out sets, it is common to test a model using *n-fold cross-validation*. On each “fold” a subset of items is chosen for the hold-out set, and different hold-out sets are selected for different folds, such that, across folds, all items appear in exactly one hold-out set. Each hold-out set provides a measure of model classification accuracy, and this is usually averaged across folds to provide a single number indicating how accurately the trained model can classify hold-out patterns. We will refer to this number as the *cross-validation accuracy* of the classifier.

MVPA algorithms, like univariate analyses, are challenged by the abundance of data provided by fMRI, and so must adopt additional assumptions about the nature of the underlying signal. In any fMRI study (as in our model) there will always be more predictors (voxels) than things predicted (stimulus items or events), producing an over-fitting problem. In such cases, there exists no unique solution to the classification problem defined by the training set. Closed-form analyses are undefined, and other model-estimation procedures will produce a classifier that perfectly fits the training data without any guarantee of finding real signal. These problems can only be addressed by constraining the analysis based on an underlying hypothesis about how signal is truly encoded in the data. As with the univariate method, these constraints systematically affect the results. Each of the remaining methods adopt different constraints to solve the over-fitting problem.

### **Searchlight concepts and assumptions.** We begin with the well-known “searchlight” approach (Kriegeskorte et al., 2006), which was formulated specifically to address the challenge of finding distributed representations in brain imaging data. The method works as follows. Instead of training a classifier using all predictors at once, a separate classifier is trained for every individual voxel location in every individual subject. For each location, all voxels within a radius of the center voxel are included as predictors in the classifier. This avoids the over-fitting problem by restricting the number of predictors included in any given classifier. The mean cross-validation accuracy for each classifier is stored in the searchlight center voxel, providing an *information map* for each subject. An univariate group-level analysis can be conducted on the information maps, similar in all respects to the analysis described in the previous section. Per the univariate assumptions, this means that each point in the accuracy map is considered independent of all others. However, within each searchlight, the effect of a given unit on the classification can differ depending upon the activations of other units in the searchlight, and these units can respond to various stimuli in quite different ways.

Thus the searchlight method relaxes assumptions about the consistency of the neural code within and across individuals, and about the independence of representational units, but retains assumptions about localization of information within and across individuals. These assumptions are summarized in Table 1, and can be contrasted with those of the univariate method. What results are observed in the model with these differing representational assumptions?

Implementation. The searchlight analysis was conducted using the SearchMight toolbox (Pereira & Botvinick, 2011) for MATLAB (Mathworks, 2013a). The input units, hidden units, and output units were treated as three anatomically separated regions so that a searchlight never encompassed units in different regions. This was accomplished by inserting empty units between the layers, and providing a mask to SearchMight to omit those units during analysis while ensuring that no searchlight spans multiple regions. Within each searchlight, a Gaussian Naive Bayes (GNB) classifier was fit to distinguish between category A and B items. Although GNB classifiers are limited in some ways (Pereira & Botvinick, 2011), the concerns do not apply to this simple and idealized case where noise is truly identically and independently distributed (iid) with uniform variance. The amount of category information in each searchlight was estimated through 6-fold cross validation; the mean cross-validation accuracy was stored at each searchlight center; and the mean accuracy over model subjects was then computed for each unit and tested to see if it differed significantly from chance. The resulting map of p-values is FDR corrected, q<0.05.

As previously, the analysis was performed on both the anatomically localized and the anatomically dispersed arrangements of units. The data were not smoothed prior to the searchlight analysis. The analysis was performed with various searchlight sizes, ranging from 3 to 28. A searchlight size of 7 or 9 should be roughly “optimal” given the size of the clusters of informative units in the localized data.

Results. Figure 5 shows the results of the searchlight analyses, for localized and dispersed model architectures, and for different searchlight sizes. The format is the same as in the preceding analysis, except the y-axis now indicates the mean classification accuracy for searchlights centered on each unit, rather than a t-value at each unit. As before, colored bars indicate units that the method identifies as statistically significant---that is, units whose surrounding searchlights show classification accuracy reliably above chance across model individuals.

**---Figure 5 about here---**

There are several points to note in these plots. First, when the hidden representations are anatomically localized, the method can do a quite good job of identifying both the systematic I/O and the systematic hidden units as important for the domain representations, though for both unit types the results vary substantially with the searchlight size. When the searchlight is small, the method reliably finds the SH units but misses most of the systematic I/O units. This happens because, as noted earlier, the SH units encode a clearer differentiation between domains, so that even if the searchlight does not encompass all 7 units there is sufficient information within it to classify stimuli above chance. The domain distinction is weaker in the systematic-I/O units, so when only a small number of such units fall within the searchlight, there is insufficient information to classify correctly. With a larger searchlight (9 units), the method does very well at finding all relevant units. When it grows too large, however, it begins to incorrectly flag irrelevant units as being important for the representation (28 units). A very large searchlight, even when centered on an uninformative (arbitrary) unit, can have a broad enough span that it encompasses other informative units. In this case the classifier will perform well by virtue of the informative units appearing in the edge of the searchlight, but the above chance result will be “stored” in the searchlight center, making it appear as though there is useful information present at that location. Thus when the signal is anatomically localized, there is a tradeoff between searchlight size and discovery of representational structure, with searchlights that are too small missing weaker signal and those that are too large incorrectly flagging arbitrary signal.

In the model case, where we know *a priori* which are the signal-bearing units, it is easy to discern the optimal searchlight size, but it is less clear how this would be determined from real brain imaging data. One might initially expect the optimal searchlight to be identifiable from the accuracy of the resulting classifiers, but Figure 5 suggests that this is not the case: the very large searchlight, which flags many irrelevant units, shows almost as good classifier performance as the optimal size at the SH units and better performance at the systematic I/O units. If we did not already know which units were important for representation in the model, it would be difficult to know which searchlight size to choose, and hence which results to believe.

The second thing to note is that the searchlight analysis does a much poorer job overall of identifying the SH units when these are anatomically dispersed (right panels of Figure 5). The poor performance arises because the precise anatomical location of the signal-carrying units is assumed to vary across individuals in this case. Within any individual model, a searchlight that includes a few of the informative units will show above-chance performance in classification, but the searchlight centers will differ across model individuals, especially when the searchlights are small. Thus the cross-subject statistical test at each location will yield a null result, leading to poor signal discovery. Larger searchlights will be more likely to contain the signal-carrying units, but also lead to poorer localization of the signal as already noted.

In sum, the method deals with the over-fitting problem by only including a small number of contiguous voxels in each classifier---an approach which assumes that useful representational structure can be localized within the searchlight radius, in the same locations across subjects. When these assumptions are met, the approach does a good job of discovering representational structure, even if the representational code (i.e., the way that individual units respond to particular stimuli) is highly variable within and across individuals. The limitations noted above arise when the assumptions are violated---when representational structure is anatomically distributed across multiple searchlights (as when searchlights are too small in the localized case), or in different ways across individuals (as in the dispersed model). Moreover, whether the assumptions are met depends, not only upon the anatomical distribution of the signal, but also upon the searchlight size, and it is not clear how the latter can be optimized for real brain imaging data.

Finally, it is worth noting that, in contrast to the univariate method, the searchlight approach does not provide information about how the contrast of interest in encoded in unit activations. Thus there is no way for the method to show, for instance, that there are some units systematically more active for A items than B items, others showing the reverse pattern, and still others that express the A/B distinction in a distributed code (the SH units). These results are summarized in Table 2.

## Regularized logistic regression for whole-brain pattern classification

Concepts. The limitations of the searchlight method arise from the relationship between a searchlight's field of view and the anatomical distribution of the underlying signal. A small searchlight provides better localization but is more likely to exclude signal-carrying units; a large searchlight is more likely to include the signal-carrying units but provides less information about where the signal really is.

An alternative approach that avoids this tradeoff is to train a pattern classifier on the whole dataset simultaneously. While many classification algorithms exist, we focus here on variants of logistic regression because they are easily interpretable, powerful, and draw upon intuitions formed through experience with linear regression. A regression model is composed of a set of weights , one for each predictor variable plus an additional intercept term, tuned to make accurate predictions about a response variable . In logistic multivariate pattern classification, the predictor variables are the voxels, and the response is a binary variable that codes class or condition label. For instance, in a contrast of conditions A and B, A events are labeled with and B events are labeled with . To generate a prediction for a given item, the logistic regression model takes the weighted sum of the estimated response over voxels and passes it through a squashing function bounded at 0 and 1:

|  |  |
| --- | --- |
|  | (1) |

where:

|  |  |
| --- | --- |
|  | (2) |

which is the model's linear response to a particular pattern of activity. Thus, is a transformation of the weighted sum of predictor values expressing the probability that y=1 given the pattern of activity for the item. Fitting a logistic regression model involves finding coefficients that minimize the discrepancy between the true labels in and the probabilities assigned by the model. This is typically measured by the logistic loss:

|  |  |
| --- | --- |
|  | (3) |

where is when and +1 when . This loss is minimized when the sign of the model's linear response is positive for items labeled and negative for items labeled.

As noted previously, the problem is that there are infinite possible solutions to the minimization when there are more predictors than items. One needs a way of deciding which among these is most likely to uncover the real signal. *Regularized* regression provides one way of doing this. Such approaches seek to jointly minimize the prediction error plus an additional cost, itself a function of the coefficients:

|  |  |
| --- | --- |
|  | (4) |

The additional penalty or *regularizer* represented by prioritizes some model solutions over others, and in this way embodies a hypothesis about the nature of the true underlying signal. The constant is a free parameter that controls the degree to which the two terms (prediction error versus minimization of the regularizer) should be weighted in the joint optimization.

We here consider two varieties of regularized logistic regression recently employed in the fMRI literature: *Lasso* (Tibshirani, 1996; Rish, Cecchi, Heuton, Baliki, & Apkarian, 2012) and *ridge regression* (Hoerl & Kennard, 2000; Riggall & Postle, 2012). Though superficially similar, the two methods embody different implicit assumptions about the nature of the underlying signal and so yield quite different results. In Lasso, the regularizer is the sum of the absolute values of the model coefficients:

|  |  |
| --- | --- |
|  | (5) |

For ridge regression, the penalty is the sum of their squared values:

|  |  |
| --- | --- |
|  | (6) |

In both cases, the optimization is convex: for a given value of , there exists a unique set of coefficients that minimize the cost and that can be efficiently discovered by gradient descent. Yet the different penalties lead to quite different solutions. To understand why, it is useful to consider how they treat sets of predictors that covary together with one another. Imagine four voxels whose responses across items are perfectly correlated, and suppose their activations are useful in predicting the condition label. In this scenario, there are many different ways of placing weights over the four voxels that will all have the same effect on the classifier output. For instance, placing a weight of on each voxel will have exactly the same effect as placing a weight of on one voxel and a weight of on the other three. Because the voxel activations are perfectly correlated, and the classifier operates on a weighted sum over voxel activations, these different weight configurations have the same effect on the model output and hence on the prediction error. The regularization penalty, however, should prefer some weight configurations over others.

If the data really are perfectly correlated, the Lasso penalty won't be any help: the sum of the absolute value of the coefficients is the same for models that place a on each unit versus models that put a 4 on one unit and zeroes on the rest. If we imagine, however, that all measurements are subject to some independent noise, the scenario is a bit different. In this case, one of the 4 units will, just by chance, covary slightly better with the category labels. In this case, the classifier can do a slightly better job of minimizing the error term by loading up all of the weight on this single voxel. Thus the joint optimization will lead to a solution where just one (or perhaps a few) of the redundant voxels are selected.

Ridge regression behaves very differently. Here the penalty scales exponentially as weights increase on a single voxel, but only linearly as weights are added across voxels. Thus the penalty is minimized by placing small weights on many voxels. In the preceding example, placing a weight of on one unit and on the remaining three leads to a total penalty of over the four units. Placing a weight of on each unit, in contrast, leads to a penalty of . Ridge regression thus prefers solutions where small weights are “spread out” over redundant predictors. In fact, as the weight approaches zero, the ridge penalty becomes vanishingly small, so with a finite number of training examples, ridge regression will always place at least a tiny weight on every predictor. In real data, of course, voxel states are never perfectly correlated nor perfectly informative about the condition label, so the behaviors of the two approaches are less easy to intuit. In general, however, it is useful to think of Lasso as minimizing prediction error with the *fewest* possible predictors (i.e., as many zero coefficients as possible), while ridge regression can be viewed as “spreading” small weights over all predictors exhibiting any systematic relationship with the category labels, without care for the number of predictors.

Table 1 summarizes the assumptions made by these approaches---all are relaxed relative to the univariate and searchlight methods. This does not mean that they are assumption-free. To the contrary, each approach entails additional assumptions about the *sparsity* and the *redundancy* of the underlying signal:

1. Sparsity: Lasso assumes the signal to be *sparse*, in that only a small proportion of voxels are involved in coding the information of interest. In this case, the best approach to finding true signal is to minimize prediction error using the smallest number of predictors possible. Ridge regression makes no sparsity assumption.
2. Redundancy: Ridge regression assumes that the signal is highly redundant, so that many voxels express essentially the same information. In this case the best approach to finding true signal is to minimize prediction error using distributions of weights that are as close to zero as possible, so that all informative predictors are included in the solution. Lasso assumes that the underlying signal is *not* highly redundant, so that different predictors carry different information.

With these assumptions, what signal do Lasso and ridge regression detect in the model?

Implementation. Logistic Lasso and ridge regression were conducted using GLMNET (Friedman, Hastie, & Tibshirani, 2010) in MATLAB (2013a). Both methods have a free parameter that controls the importance of the regularization penalty relative to the prediction error, leading to greater sparsity in Lasso and more severe weight shrinkage in ridge regression. The analysis thus proceeded in two steps: one to estimate a useful for each subject, and a second to fit a model at the estimated and evaluate it on a hold-out set. The data for each model subject was first divided into 6 equal parts, each containing the same number of category A and B items. One part was set aside and the remaining 5 were passed to a function that conducted a 5-fold cross validation accuracy test at 100 values of . The function returns the producing the highest cross-validation accuracy, which is subsequently used to fit a model to all 5 parts of the data. The resulting model was then assessed on the original hold-out set (the 6th part). This procedure was carried out separately for all 10 model subjects, in both localized and anatomically dispersed model variants, for both Lasso and ridge regression. For each model subject, each method returns a vector of coefficients that indicates how the classifier interprets each unit's activation in generating a predicted class label. To understand which units contribute to the representation of interest and how these units encode information, the coefficients must be interpreted. A key difference between methods lies in the ease of interpretation. We will therefore consider results from the two methods separately, before contrasting them.

Results for Lasso. For Lasso, interpretation of the classifier coefficients is straightforward: the method places zero weights on as many predictors as possible, so any unit receiving a non-zero weight can be viewed as having been “selected” by the classifier as important. If the classifier shows above-chance cross-validation accuracy, we can be certain that it has successfully identified signal-carrying units. In contrast to the preceding methods, there is no statistical test performed on a null hypothesis at each unit. Instead, any selected unit in a given model individual can be viewed as “significant” for the representation because, if it could be discarded without affecting classifier performance, Lasso would have done so. In this sense, for each subject, the method can be viewed as finding the *smallest sufficient set* for classification. The central questions then are (1) how well does the selected set pick out the signal-carrying voxels in I/O and hidden layers, (2) do the classifier weights indicate differences in how information is encoded for different voxel sets and (3) do the results differ for localized versus dispersed model variants?

**---Figure 6 about here---**

Figure 6 shows how often Lasso selected each unit across the 10 model subjects for localized versus dispersed cases. To get a sense of which units were selected more often than expected by chance, we took the overall proportion of units selected across subjects as a base probability for conducting a binomial test at each unit. Colored bars indicate units that were selected more frequently than expected if Lasso was choosing at random with this base rate, without correction for multiple comparisons. From this plot, the approach does a fairly good job of identifying the SH units, reliably tagging 5 of the 7 units (71%). The approach did less well discovering the systematic I/O units, reliably identifying only 6/36 (17%). This difference reflects the fidelity of the representations coded across different unit sets: as already noted, the 7 SH units encode the cleanest representation of the domain distinction, and so are more likely to be included in the smallest sufficient set for any individual. Also, note that the results are identical for localized versus dispersed cases. Since Lasso is conducted separately for each individual and is blind to anatomical structure, the results are literally identical regardless of how the units are spatially arranged.

This summary plot is misleading in one sense, however, since it applies an aggregate statistical test across model individuals to assess which units are reliably discovered. Such a test would not be possible with real data, since it would not be clear which voxels should be “lined up” across subjects to compute the binomial probabilities. The virtue of Lasso (and ridge regression) is that they are essentially single-subject analyses, and so are freed from assumptions about consistency in location and coding across individuals. What we really wish to know is how accurately the solution picks out the units of interest *for each individual model*. For every model individual, from the binary classification of selected versus unselected units, we can compute two numbers that jointly describe how well the solution identifies the important units. Specifically, we compute the *hit rate*, which is the proportion of actual signal-carrying units identified by the algorithm, and the *precision*, which indicates what proportion of the selected units are true signal-carrying voxels. Moreover, these figures can be tallied for just the I/O units, just the hidden units, and for the whole network, to provide an indication of how well the method singles out informative units in these different sets.

**---Figure 7 about here---**

Figure 7 shows the mean of these figures across model individuals for Lasso (and other methods). The general pattern is clear: precision is relatively high, indicating that most of the units Lasso identifies are indeed signal-carrying units. Lasso is sub-optimal, however, in the hit rate: for the hidden layer, about half the important units are missed, while the great majority of signal-carrying units are missed in the I/O layer. Thus if Lasso selects a unit, one can have confidence that it does carry useful information, but one cannot have confidence that it has discovered all the useful elements.

Finally, we can ask how well Lasso uncovers differences in the representational code. The red-blue spectrum of the colored bars in Figure 6 indicates the frequency with which each unit receives a positive weight across model subjects. Red and blue colors indicate that a unit's activation receives the same interpretation across model subjects, while shades in between indicate that the interpretation varies. The Figure shows that, where Lasso does identify systematic I/O units, it also reveals the correct code: all units are red or blue. There are so few units identified, however, it is difficult to “see” the systematic layout of these responses. In the SH layer, Lasso correctly indicates that code can vary across individuals for some units, though it also appears to show consistent category-selective responses for some units. These differences arise because Lasso does not succeed in selecting all SH units in every model individual. Instead, each unit is identified in about half of the individuals. When the algorithm selects a unit in a small set of participants, all of whom happen to have acquired the same code, the selected unit appears to show a selective code. These results are summarized in Table 2.

Results for ridge regression. The ridge regression classifier showed marginally better cross-validation accuracy (0.65 compared to 0.6 for Lasso), but with a very different distribution of weights. In fact, ridge regression placed a non-zero weight on every unit---effectively using the whole pattern of activation across all units in the network. Consequently it is difficult to know which predictors are playing an important role in the classifier behavior and which are not. Weight size (i.e., absolute value of a weight) provides one indicator of predictor importance, since the regularization penalty tries to keep weights as close to zero as possible. Any predictor receiving a weight that deviates strongly from zero must, therefore, be important for reducing prediction error. But this relationship is not perfectly transparent. Consider the case where a single unit carries important information for classifying one subset of items, while ten highly redundant units all carry information important for classifying another subset. In some sense all 11 units are equally informative for the classifier, but ridge regression will place a large weight on the singleton unit and many small weights across the ten redundant units. That is, the weight size under ridge is sensitive to both the informativeness of the unit activation and its redundancy with other units. Highly redundant units can receive quite small weights even if they carry useful information. For these reasons it is not clear, in the model and more so in real data, just how strong or weak a weight must be to “count” as having been selected by the classifier.

**---Figure 8 about here---**

Figure 8 illustrates these points by showing the mean, over model subjects, of the absolute value of the classifier weight at each unit. It is clear that signal-carrying units receive somewhat stronger weights than the arbitrary units overall. It is also clear that the SH units receive stronger weights on average than do the systematic I/O units, reflecting their greater utility in reducing prediction error. Intuitively, one wants to draw a threshold below which units are classified as irrelevant, but it is not clear how the threshold is to be selected. The differences in weight magnitude are not large, and there is no a-priori basis for deciding how small a weight should be in order to conclude that it is not useful. Yet the conclusions one draws about where the signal is encoded can vary fairly dramatically depending upon this decision. The intensity of the shading in Figure 8 indicates which units would be “selected” under different thresholding policies. With a very strict policy (discarding 66% of the units), the representation would appear to reside mainly within the SH units. With a more lax policy (discarding 33% of the units) it would appear to be very broadly distributed over many units.

As with Lasso, the aggregate plot is somewhat misleading, since the different selection policies operate on mean coefficient values that could not be calculated in real data unless representations were localized identically across individuals. We therefore conducted the same analysis of hit rates and precision values across model individuals, adopting three different policies for discarding small weight values. In the *lax* policy, the 21 units (20%) with the weakest classifier weights were deemed unselected; in the *moderate* policy, half of the weights were discarded; and in the *aggressive* policy, only the 21 units with the strongest classifier weights were retained. For each policy we computed hit rates and precision, for I/O units, hidden units, and all units. The results are included in Figure 7. When the policy is lax, the pattern is opposite to that observed in Lasso: relatively high hit rates but low precision for all unit subsets, indicating that the method has incorrectly selected many arbitrary units. When the policy is aggressive, the pattern is similar to Lasso: low hit rates but relatively high precision, especially for the SH units. Thus the accuracy with which the classifier weights pick out the signal-carrying units varies dramatically depending on the arbitrary selection of a weight threshold. In the model we can, in principle, discover an optimal thresholding policy---one that maximizes hit rate and precision---but only because we already know the ground truth. With real data, where the number of predictors is much larger, the representational structure likely to be much more complex, and with no knowledge of the ground truth, it is not clear how the set of weights discovered by ridge regression might be used to discover where the useful signal is coming from.

Finally, does ridge regression provide useful information about the different nature of the representational code at different units? As previously, the hue of the colored bars in Figure 8 indicate the frequency with which a unit receives a positive weight across model subjects. Both the independent coding of domain in I/O units and the variable nature of the code across subjects at the SH units come across fairly clearly. Thus the method does a reasonable job of highlighting differences in the representational code across these units subsets. The chief problem with the approach is the difficulty it poses in understanding which units are contributing meaningfully to the classification.

These results are summarized in Table 2.

## Sparse-overlapping-sets (SOS) Lasso

Concepts and assumptions. The multivariate methods we have considered so far lie at two poles with regard to their assumptions. The searchlight method assumes both localization of signal within individuals and consistency of localization across individuals. Regularized regression discards both assumptions, allowing for the possibility that representations of interest are arbitrarily situated within individuals, in completely different ways across individuals. From the PDP view of representation, the former assumptions seem too restrictive, while the latter assumptions seem too loose. If we believe that all human beings share the same gross neuroanatomical structure, then it seems reasonable to suppose *some degree* of localization within individuals and some degree of consistency in location across individuals. Within individuals we might assume that, if a given unit is important for representation, there is a good chance that some other units in the general vicinity are also important. Such an assumption could be adopted without further requiring that *all* neighboring units are important (univariate assumption 1), or that *only* neighboring units contain sufficient information for the representation (searchlight assumption 1). Likewise we might assume that, if a set of voxels contribute to a representation in individual A, then signal-carrying voxels are likely to be found in the same general anatomical vicinity in individual B. Such an assumption could be adopted without requiring that the location of the interesting information is very precisely aligned, or that the location contains useful information in the great majority of individuals (assumption 3 for univariate and searchlight).

With these intuitions in mind, the final method we consider is the Sparse Overlapping Sets (SOS) Lasso (Rao, Cox, Nowak, and Rogers, 2013). SOS Lasso is a variant of Lasso that allows for evaluation of a range of hypotheses regarding localization within and across individuals. It adopts a central and intuitive assumption from *multi-task learning* (Caruana, 1997): if a set of learning tasks are related, then their solutions will also be related. In this case, each “task” is to find the neural activation pattern associated with a particular mental state in a single individual. In contrast to Lasso, we assume that the solutions across individuals are related: finding useful information in one individual gives us clues about where to look in another. The loose assumptions about localization within and across individuals can be built into a single optimization function that considers all subjects at once.

To achieve this, the voxels in the datasets from every individual are assigned to *sets*. Within an individual, voxels belonging to a given set are related to one another in some way--for instance, if the sets are defined by anatomical proximity, voxels within a set will all be near one another. Across individuals sets are aligned, so that voxels in *Set 1* for individual 1 are related to voxels in *Set 1* of individuals 2 through *n*. If sets are defined by anatomical proximity, then voxels in Set 1 will be localized similarly across subjects. The optimization then operates over all subject datasets simultaneously, minimizing an objective function that prefers solutions in which selected voxels belong to as few sets as possible. The end result is a solution that is unique for each dataset (i.e., each individual), but with the selected voxels tending to belong to the same sets within and across individuals.

Before considering the objective function itself, it is useful to consider how voxels are assigned to sets. Such assignments are made *a priori*, and can in principle be guided by any hypothesis about how voxel sets are best grouped. For instance, sets might be determined by an anatomical segmentation, a previous analysis, a literature review, or any other prior knowledge. Importantly, however, set assignment can also be done in a relatively theory-neutral way: a volume can be split into a collection of *overlapping* sets of fixed size and degree of overlap, with each voxel belonging to multiple overlapping sets. On this approach, the voxels for each individual are first projected into a common anatomical reference space (without blurring). The space is divided into a 3-dimensional grid and, for each grid-point, all voxels within radius are assigned to a given set. Each set thus contains a group of voxels in roughly the same anatomical location across subjects; each voxel belongs to multiple sets; and the sets overlap in the voxels they contain. The sets are analogous to searchlights, but rather than training separate classifiers for each set in each subject, a regularized logistic classifier is trained on all the data together. Because the optimization prefers solutions in which voxels belong to a small number of sets, such an approach will automatically select the particular grouping of voxels within and across individuals that best minimizes the objective. But, because the solutions are unique to each individual, there is no need for all voxels within a set to be selected or to receive similar weightings within or across individuals.

With that conceptual grounding, we can consider the SOS Lasso regularization penalty:

|  |  |
| --- | --- |
|  | (7) |

The function has two terms, and , which are embedded in a loop that sums over sets. The second term is simply the Lasso penalty from Equation 5, computed separately for each set and summed over all sets. The first term, which we will refer to as the *set penalty*, takes the root of the squared coefficients summed over units within each set. is a mixing parameter that determines the weight given to each of these two terms. When , the set penalty is not applied and SOS Lasso effectively reduces to Lasso. When is non-zero, however, the penalty behaves differently, preferring solutions where the non-zero coefficients (i.e., the selected voxels) all appear in the same overlapping sets---that is, when they tend to be in similar anatomical locations.

**---Figure 9 about here---**

To understand how this preference is realized, we can compute the penalty associated with two competing solutions in a very simple case, diagrammed in Figure 9. Consider a 1-dimensional brain with 3 voxels. Voxels 1 and 2 are assigned to *Set 1*, and voxels 2 and 3 are assigned to *Set 2*. In the first solution, let voxels 1 and 3 be assigned a value of , and voxel 2 a value of . The set penalty evaluates simply to , because the voxels are in different sets. In a second solution, let voxels 1 and 2 be assigned a value of and voxel 3 a value of . This time, the penalty evaluates to , because the voxels are in the same set. Because the set penalty takes the root of the sum of squares across voxels in the set, the assignment of a value to 2 of the 3 voxels produces a larger cost when the coefficients are placed on voxels from different sets.

There is a wrinkle here, of course: voxel 2 also exists in *Set 2*. SOS Lasso avoids double counting by applying each weight only where it is least expensive. In the current case, if we consider the weight assigned to voxel 2 as part of *Set 2*, it “costs” penalty unit. If we consider it as part of *Set 1* it only costs penalty units. In principle the total weight on the voxel could be apportioned across its corresponding sets in any manner; however allocation of the voxel's weight to an otherwise empty group incurs a greater overall penalty, so the optimization ends up placing all of the voxel's weights within sets that also contain other non-zero voxels. In this way the optimization “knows” to account for voxel 2 as belonging to *Set 1* rather than *Set 2*, and does not redundantly penalize the same weight. The further consequence is that the optimization automatically finds sets that contain many signal-carrying voxels if these are present in the data.

Finally, it is important to note that both the *set penalty* and the Lasso penalty increase when the magnitude of any weight is increased. This means that, just as with Lasso, solutions that include fewer voxels overall will tend to be preferred. Solutions will be sparse, even within groups. Sets are not selected themselves, but rather voxels are selected from within sets and it is less costly to select multiple voxels from the same set than from different sets.

In sum, the optimization prefers solutions that (a) fit the training data, (b) include units that are anatomically situated in similar sets, and (c) include a relatively small proportion of the units overall. As with Lasso and ridge regression, there is a tuning parameter that controls the weight given to the regularization penalty versus the error term; but there is also a second tuning parameter ( in the above equation) that controls the relative weighting given to the set penalty versus the Lasso penalty. A mathematical analysis of SOS Lasso and some applications of its use were recently described by Rao et al. (2013) for linear regression and by Rao, Nowak, Cox, and Rogers (2014) for logistic regression. Again, the representational assumptions of SOS Lasso are summarized relative to those of the other methods in Table 1.

By assuming that signal is potentially both structured and sparse, how does this approach fare in analysis of model data?

Implementation. The SOS Lasso analysis was implemented using custom code built on top of MALSAR (Zhou, Chen, & Ye, 2011) for MATLAB. The data were divided into overlapping groups based on “anatomical” proximity. Each set included 5 units, and the groups overlapped with each other by 2 units. The set size was kept smaller than the smallest expected cluster of informative units in the localized data, but was otherwise arbitrary. SOS Lasso has two free parameters, one controlling sparsity at the set level and one controlling overall sparsity. As in the previous analysis, these parameter values were selected through an internal 5-fold cross-validation process, then a final model was trained with the best parameters and tested on a sixth hold-out set. Note that SOS Lasso is trained on all model subjects simultaneously, but produces a unique solution for each subject. Thus the results were analyzed in the same way as the Lasso and ridge regression results.

Results. The first point of note is that the SOS Lasso classifier showed a mean cross-validation accuracy across subjects of 0.82 in both the localized and anatomically dispersed model variants---much higher than searchlight, Lasso, and ridge regression, all of which showed accuracies between 0.6--0.65. This stark difference suggests that SOS Lasso is able to exploit cross-subject consistency in the location of useful units to better find the informative signal in individual models.

**---Figure 10 about here---**

Figure 10 shows, for each unit, how frequently it was identified as important across model individuals. As with Lasso, the base-rate of unit selection was used to conduct a binomial test across model individuals, to indicate which units were selected more often than expected by chance. The method clearly excels at identifying the SH units. When these are localized, all units are reliably identified, but even when they are anatomically dispersed, the method reliably finds 6 of the 7 units. Arbitrary units are almost never flagged as important for representation. Compared to Lasso, the method also does a slightly better job identifying the systematic I/O units, reliably identifying 11 of the 36 units (31%) in both cases.

The rightmost panel of Figure 7 shows the mean precision and hit rate across individuals in the anatomically localized and dispersed models. Considering all units, SOS Lasso maintains the same precision as Lasso, but with a substantially better hit rate. With localized representations, this improvement was observed for both hidden and I/O units. When hidden representations were anatomically dispersed, the precision declined slightly but the hit rate still improved markedly. In the dispersed case the precision declines when the method sometimes selects arbitrary units belonging to groups containing signal-carrying units. An arbitrary unit contained in a selected set and happening to spuriously covary with the class label may itself be selected. Despite such inclusions, however, the overall gain in hit rate with minimal loss of precision, together with the greatly improved classifier cross-validation accuracy, suggest that SOS Lasso is doing a substantially better job of finding useful representational structure.

The advantages of SOS Lasso are clear when representations are anatomically localized, but what accounts for the improved hit rate when they are dispersed? The first observation is that, though signal-carrying voxels do not line up across subjects, there exist some sets that happen to contain many signal-carrying voxels across subjects, while others contain mostly arbitrary voxels. Because selecting voxels from many sets is more expensive than selecting the same number of voxels from fewer sets, SOS Lasso strives to select voxels from the smallest number of sets possible. If this were the only reason, however, then the searchlight method should have also succeeded. The second observation is that all of the selected voxels across groups contribute to a single classifier, just as in Lasso and ridge regression. Thus SOS Lasso can perform well even when the useful information is distributed over multiple sets. In general SOS Lasso builds the best sparse whole-brain classifier possible while keeping solutions for different subjects as similar as possible, where similarity is defined by membership in the same set.

It is worth noting here that the SOS Lasso is a generalized optimization within which Lasso itself is a special case. As noted earlier, when the parameter controlling the weight on the group sparsity penalty is set to zero, the regularizer reduces to the standard Lasso penalty. This means that the parameterization of SOS Lasso itself indicates the extent to which there exists useful cross-subject consistency in signal location. If no such consistency exists, the classifier cross-validation accuracy will not exceed that of Lasso alone, and the best estimate for the group-sparsity parameter will be near zero. For real data, this suggests a way of testing whether there exists cross-subject consistency in information localization. The SOS Lasso classifier will only perform better than the Lasso classifier if such consistency exists.

Finally, with regard to laying bare the representational code, SOS Lasso performs similarly to Lasso: Where units are reliably discovered, the nature of the code is clear from the weight values (colored bars in Figure 10), though these patterns are somewhat noisy when the units are only discovered in a small number of model individuals.

These results provide answers to our core questions, summarized in Table 2.

**--- Table 2 about here---**

# Application to real data

Before turning to a broader consideration of the implications of our results, it is useful to consider whether the methods of interest yield appreciably different results when applied to real fMRI data. If they don't, the discussion to this point is essentially moot. For this reason, we compared the results of the four different multivariate methods when applied to the publicly-released star-plus dataset (Mitchell et al., 2004). In each trial of this study, participants viewed a visual display showing simple shapes in a particular spatial configuration, and also read a sentence describing shapes in a particular spatial arrangement (e.g. “The star is above the triangle”). The task was to decide whether the verbal statement was true of the visual display. The two stimulus types (sentence versus display) were counterbalanced for order. Participants performed the task for 20 minutes while their brains were scanned with fMRI. Only a portion of the brain was scanned, allowing high temporal resolution: activity at each voxel was measured once every 500ms for a total of 2160 measurements per voxel. Measurements taken between trials were discarded, and remaining measurements were labeled as arising either from the sentence-processing half of a trial or from the visual-display half of the trial. The goal of the study was to learn a classifier that could successfully decode, from the BOLD response taken at a given timepoint, whether the subject was processing a sentence or a picture. To that end, the authors grouped all voxels for each subject into 25 anatomically-defined regions. Of these, they selected 7 regions hypothesized *a priori* to encode information relevant to the classification task. BOLD signal was averaged over all voxels within a region, producing 7 mean BOLD values for each subject at each time point. The authors then trained a support-vector machine classifier on data from 13 subjects. This approach led to cross-validation accuracy of 89%. More interestingly, the authors trained a classifier on data from 12 subjects (as if they were one large dataset) and tested on the thirteenth. The mean accuracy, after holding out each subject once, was 75%, indicating some consistency in how and where the useful information is encoded across individuals.

The publicly-released star-plus dataset is limited insofar as it contains functional data from only 6 participants, without accompanying anatomicals. Thus the data can be aligned only very roughly across subjects, and there is insufficient power to conduct cross-subject statistical tests. Importantly, however, the original work demonstrates that there exists cross-subject consistency in the data that can be exploited by multivariate pattern classifiers. Thus we can be certain that there exist interesting relationships across subjects in where and how information relevant to the task is encoded.

Our aim in this analysis is to assess whether searchlight, Lasso, ridge regression, and SOS Lasso yield qualitatively similar or different results when applied to these data. Note that, in contrast to the original work, none of these methods involves pre-selection of regions of interest or data reduction via averaging. Instead, each method is applied to the data in their raw form. To assess whether a method has discovered some useful representational structure, we first consider whether it shows better than chance cross-validation accuracy. The central questions then are (1) whether the methods identify similar or different voxel subsets, (2) whether they suggest similar or different conclusions about how information of interest is coded in voxel activations and (3) whether it is possible to adjudicate the validity of the different results.

## Implementation details

The analyses of the “star-plus” dataset are simply scaled up versions of the simulated analyses described above. Thus, searchlight analysis was conducted using the SearchMight package, ridge regression and Lasso were applied using GLMNET, and SOS Lasso was implemented by incorporating the new loss function into tools provided in the MALSAR package, all of which was done in MATLAB.

All classifiers were trained on data within 8s of each stimulus onset. Time points that were more than 8s from a stimulus onset were dropped. Each retained time point was labeled as an example of a sentence or a picture based on the most recently occurring stimulus type. The goal was to classify every retained time point.

Classifier accuracy was determined via 10-fold cross-validation. Examples that belonged to the same trial were always assigned to the same cross-validation fold. This ensured that, if there are any intra-trial relationships between the examples, they did not contaminate the independence of the training and hold-out sets. For searchlight, whether the mean cross-validation accuracy for each searchlight differed from chance was determined at the individual level, controlling the FDR with . We then aggregated data across subjects by counting, at each searchlight center, how many subjects individually showed greater than chance classification.

## Results

**---Figure 11 about here---**

All methods exhibited significant above-chance cross-validation accuracy, indicating that they all discover some reliable structure. Figure 11 presents the mean cross-validation accuracy of each method along with aggregate images of the brain regions identified as encoding useful structure by the different methods within one representative slice of the brain. Identified voxels are aggregated across the Z plane and projected on a single horizontal section taken through the middle of the temporal lobe, with anterior regions toward the top.

For searchlight, voxel coloring indicates the proportion of participants for whom the surrounding searchlight led to reliable above-chance classification. Though the map is not statistically thresholded in the usual manner (owing to the previously-noted limitations of the dataset), the color intensity gives a sense of what such a map would look like. The signal appears to be highly distributed all throughout the measured regions, similar to the model results obtained when the searchlight was too large. There are no regions where signal is obviously absent.

In the remaining panels, red indicates voxels where high BOLD was associated with picture processing, blue indicates voxels where high BOLD was associated with sentence processing, and purple indicates voxels with different interpretations across individuals. For Lasso, selected voxels are sparsely distributed throughout the measured regions, with no discernable spatial topography apart from a denser clustering of blue voxels toward the occipital cortex. Most points are either red or blue because there was relatively little overlap in the selected voxels across subjects. The ridge regression plots show results at two levels of weight thresholding (10% and 90%). Here the voxel hue indicates the proportion of positive weights across individuals, while the intensity indicates the mean absolute value of the voxel weight across individuals. In both cases, the results are a “purple haze”: almost everything is selected in some participant, and there is little apparent coherence across participants in how a given voxel encodes information. Finally, the SOS Lasso plot identifies fewer voxels overall, but these tend to be loosely localized within three regions: the occipital cortex, and left and right anterior temporal cortices. Within these regions, the direction of the classifier weights appears largely intermixed, consistent with the PDP hypothesis that the nature of the neural code can be highly variable both within and across individuals.

From these images, it is clear that the different methods do indeed yield very different conclusions about how the relevant signal is encoded. Which view are we to believe? Cross-validation accuracy of the different classifiers might provide one clue: presumably solutions that have done a better job identifying important components of the representation should show better cross-validation performance. We therefore compared the mean cross-validation accuracy across the different methods. For searchlight, this was measured in each participant by taking the mean accuracy over all searchlights individually showing reliable above-chance classification. The other methods each directly yield a single measure of accuracy for each participant. Means and standard errors of these figures across participants are shown in the top right panel of Figure 11. SOS Lasso shows significantly better performance than the other three methods, suggesting that it has done a better job of picking out useful representational structure in each individual.

A second way of assessing the different solutions is to compare them to *a priori* expectations. In this case, recall that the authors of the star-plus study originally identified 7 regions of interest where they expected functional activations to carry important information. Thus we can inquire how well the solution for each method aligns with these expectations, by computing what proportion of the identified voxels fall within these 7 ROIs. For SOS Lasso, 65% of the identified voxels on average across subjects fell into the expert-defined ROIs---significantly more than all other methods (44% for LASSO, 26% for both variants of ridge regression, and 22% for searchlight, for within-subject contrast to SOS Lasso for all methods).

# General Discussion

We began by noting that an important theoretical approach in cognitive science, the PDP approach, adopts specific claims about the nature of mental representations that, despite being quite useful for understanding many aspects of cognition, have been difficult to empirically test in functional brain imaging studies. PDP suggests that mental representations are patterns of activation distributed over neural populations, with information encoded, not in the activity of individual cortical units taken independently, but by the complete pattern coded over a representational ensemble. It further suggests that the response of any given element in an ensemble may vary arbitrarily across individuals, even if the ensemble jointly codes the same structure across individuals; that elements of an ensemble need not all be anatomical neighbors within an individual; and that their location across individuals, while not completely random, may be subject to at least some variability. These assumptions about representation, paired with the observation that functional brain imaging technologies yield up vast amounts of quite noisy data, present serious challenges to discovery of the neural bases of mental representation.

We have now reviewed five different methods for analyzing brain imaging data, considered their underlying assumptions, and assessed their ability to discover distributed representations of the kind PDP assumes. From this analysis it is clear that very different results arise depending upon the statistical method employed or, in some cases, upon parameterization of the method in question. Each method succeeds best when the implicit assumptions it adopts are met in the data. Thus univariate contrast successfully identifies the systematic I/O units, which conform to the assumptions listed in the corresponding column of 1. The SH units, despite encoding cleaner domain representations, violate all of these assumptions and so are completely missed. Searchlight does well at detecting both systematic I/O and SH units, but only if the useful information is contained within the radius of a relatively small searchlight and is localized in the same way across individuals. Lasso performs modestly well at discovering SH units, but in assuming no consistency across model individuals and no redundancy within individuals, becomes highly susceptible to noise and so misses many important units. Ridge regression, in assuming highly redundant signal without care for overall sparsity, spreads weights over all units, making the solution hard to decipher even if the classifier performs well.

SOS Lasso succeeds best at finding distributed internal representations because it adopts assumptions that align well with PDP. Like other MVPA methods, no consistency in coding is assumed across individuals, allowing for differences in how the same representational structure is expressed over hidden units. In contrast to Lasso and ridge regression, representational units are assumed to be localized loosely within individuals, and with loose consistency of location across individuals, allowing for some degree of similarity in the global architecture. In contrast to univariate and searchlight methods, however, there is no assumption that all units within a given locale code the same representation, or that an informative region must contain sufficient information for a representation. This allows for the possibility that representations may be coded across multiple anatomically distal regions, and that the specific anatomical arrangement of the interesting elements can vary across individuals. The assumption that the features will be in roughly similar anatomical neighborhoods overcomes the limitations of Lasso, because the optimization will prefer units that may individually carry less direct information about the class label if these happen to reside near other informative units within or across individuals. Finally, the representation is assumed to be sparse overall, reflecting the notion that the representational elements of interest will typically be buried within a large system, with many measured components likely to be subserving unrelated functions. With these assumptions implemented in a whole-brain optimization, the method succeeds fairly well in finding the distributed representational structure that is central to cognition under the PDP view, but most difficult to find with other methods.

Application of the same methods to the star-plus dataset further suggests that, in this case at least, neural representations have properties similar to those assumed by PDP. Specifically, the patterns of activation discovered by SOS Lasso were distributed over three different coarse anatomical regions, but with different subsets of units within each region selected for different individuals, and with different representational codes apparent across voxels within a given region. With different classifiers fit to each subject under these constraints, SOS Lasso performed significantly better than other methods at classifying hold-out images, suggesting that its assumptions led to discovery of more reliable signal within each individual. In line with this view, the features it selected were significantly more likely to lie within the regions of interest shown in the original work to carry important information about the distinction of interest.

With these results, there are two broader questions we wish to consider.

## Why bother with these complicated methods?

The skeptic might wonder about the point of these analyses. Functional brain imaging has proceeded just fine for three decades now based mainly on the localizationist univariate assumptions stated earlier. Many important findings have been so discovered, and indeed, many researchers may consider the central goal of cognitive neuroscience to be the discovery of which cognitive functions are associated with which brain regions. In fact, we suspect that this view is so firmly entrenched that some of the alternative possibilities considered earlier---that representations are not always anatomically localized within a region, or that the same regions may behave quite differently across individuals---might seem far-fetched to some. Put bluntly, the critic may assert that the PDP view of mental representations is just wrong.

As a response to this point, we gesture toward the already large and rapidly growing literature on multivariate pattern analysis of functional brain imaging data, whose surface we have barely scratched. There exist many other approaches beyond those we have studied here, but by and large, they are all painting a quite different picture of neural representation and processing than suggested by the classical localizationist view. One observation is that, in many different tasks, information appears to be more widely distributed in the brain than previously thought. For instance, Mitchell et al. (2008) trained a generative model to predict whole-brain responses to nouns when given a feature-based description of the noun meaning, and showed that, across the entire brain, words with more similar meanings evoked more similar overall patterns. Rish et al. (2012) trained a variant of regularized regression similar to those explored here to predict the perceived magnitude of a stimulus from the neural response it evoked. On each iteration, she identified 100 voxels useful for this prediction task, then removed the voxels and retrained the classifier. Remarkably, she found above-chance prediction over many such iterations, retraining on an increasingly small set of predictors. The implication appears to be that, in this particular task, stimulus magnitude is reflected in the activation value of many different voxels throughout the brain. Kriegeskorte et al. (2008) has used unsupervised multidimensional scaling methods to show that the responses evoked by visual stimuli throughout ventral temporal cortex in both humans and non-human primates express coarse conceptual distinctions amongst object categories. These are just a few examples of a broad trend in the literature; many more were recently reviewed by Cox et al. (2014).

A second observation is that the information contained in distributed patterns over voxels is not mirrored by the overall magnitude of activation observed within a given cortical region. That is, univariate contrast maps and information maps often paint very different pictures of where the interesting signal is, across several studies. This point was most famously made by Haxby et al. (2001) in his MVPA analysis of the ventral temporal cortex, in which he showed that activation patterns in the putative fusiform face area (FFA) were sufficient to decode places from objects, while faces could be discriminated from objects and places from voxels outside the FFA. In a more recent example, Riggall and Postle (2012) conducted both univariate and multivariate (ridge regression) analyses of activation patterns elicited in a working memory task where participants “held in mind” the speed or direction of motion in a moving-dots display. Univariate analyses showed elevated delay-period activation of voxels in the prefrontal cortex, in keeping with many working-memory studies. From this activation, however, it was impossible to tell what kind of information (speed or direction) was held in memory. Voxels in different parts of visual cortex did not show elevated delay-period activity in the univariate analysis, but did reliably code information about the speed and direction of motion. Such studies illustrate that different regions can carry completely different kinds of task-related activation, each of which might be invisible depending upon the analysis method.

Third, there is increasing evidence, even from univariate methods, for substantial variability across participants in how information is localized. As one example, Feredoes, Tononi, and Postle (2007) conducted univariate analyses of delay-period activation in a working memory task, at both a group level and in individual subjects. In keeping with classic findings, the group-level analysis revealed reliable delay-period activation in prefrontal cortex. The effect size in individual subjects was, however, quite small---the effect was reliable because, like the systematic I/O units in our simulation, it was localized consistently across subjects. Within individual participants, much larger effects were observed in parietal cortex, but because these varied in location across subjects they were not identified in the group level analysis. Importantly, the authors subsequently applied transcranial magnetic stimulation (TMA) to either the common prefrontal region identified in the group analysis or the individualized parietal-lobe peak while participants performed the same working memory task. Application to the individual, variably localized parietal lobe region produced a greater effect on task performance than did application to the shared prefrontal hotspot, suggesting that the individualized results provided a better indication of which regions were importantly contributing to the task.

Together, these empirical findings suggest that, in many cognitive domains, the PDP assumptions may not be far off the mark, notwithstanding the many important findings that have arisen from application of univariate methods. Moreover, regardless of one's stance on the validity of these assumptions specifically, the evidence clearly indicates that different statistical methods can lead to very different conclusions about how the brain represents and processes information. This observation leads to the second question we wish to consider.

## Which is the best method to use?

We pose this question somewhat facetiously---it is not our intention to advocate for one method to the exclusion of others! Rather, the point we wish to emphasize is that each statistical method is closely associated with an implicit hypothesis about what matters in neural signal. The question of what matters---what “makes” a pattern of activation over neurons a mental representation---is itself a central question, maybe *the* central question, of cognitive neuroscience. Each of the approaches we have considered begins with an implicit hypothesis about the answer to this question. The univariate contrast method begins with the hypothesis that consistency in location and neural response across individuals is what really matters to the discovery of representation. Searchlight begins with the hypothesis that what matters is the similarity of evoked responses over neural populations within a particular contiguous region of cortex. Lasso begins with the hypothesis that representations are sparse, are not highly redundant, and can be localized any which way. Ridge regression begins with the hypothesis that representations *are* highly redundant, but still can be localized any which way.

Like any hypothesis, each of these is potentially useful in guiding discovery, but is also subject to empirical assessment. From this point of view, brain imaging might best proceed, not by choosing the “best” method, but by comparing and contrasting the results obtained across different methods. With a good understanding of the underlying assumptions each method adopts, and consequently of its blind spots, one may arrive at a better understanding of the neural code than any method individually provides. In the current work, for instance, the univariate contrast method excelled at identifying systematic I/O units but failed to find SH units. SOS Lasso showed the reverse pattern. The juxtaposition of the results provides a fuller picture of how the network represents domain structure than does either method on its own. One could imagine conducting a univariate contrast analysis, masking out all voxels that show reliable effects, then conducting a whole-brain multivariate analysis on the remaining voxels to assess if there exists a distributed code. Likewise we have noted that SOS Lasso will only out-perform individual Lasso if there is some cross-subject consistency in how representations are anatomically situated. Thus the contrast of classifier performance for Lasso and SOS Lasso on the same data provides a direct indication of whether such consistency exists. In general, it is not difficult to imagine how the different methods could be used in combination to better understand the neural basis of mental representations.

It is also worth noting, however, that the hypotheses underlying these different methods, like any hypothesis, must also play a coherent role within a broader theoretical account of the mechanisms that support the behavior of interest. In the case of the univariate contrast method, for instance, it is not generally sufficient simply to state that what matters is consistency in location and neural response across subjects. One wants some broader explanation as to *why* cross-subject consistency is an important indicator of the underlying representation. This is where we feel the connection to the PDP framework for cognition offers some real utility. The representational assumptions the approach adopts are not stipulated arbitrarily. They arise from a coherent set of principles stemming from neuroscience, computer science, and cognitive psychology that have proven useful for understanding a very broad range of behaviors in developing, healthy, and disordered populations. The hypothesis that mental representations are distributed in the particular way we have explored is, in this sense, deeply theoretically motivated. Statistical methods that begin with this hypothesis thus have a built-in theoretical justification that other methods may lack. What has been lacking, and what the new generation of statistical methods---including the methods reviewed here and other approaches beginning to emerge in the literature---is a good understanding of how to look.

# References

Allport, D. A. (1985). Distributed memory, modular systems and dysphasia. In S. K. Newman & R. Epstein (Eds.), *Current perspectives in dysphasia* (p. 207-244). Edinburgh: Churchill Livingstone.

Bates, D. (2013). *Linear mixed model implementation in lme4.* University of Wisconsin. Botvinick, M. M., & Plaut, D. C. (2004). Doing without schema hierarchies: A recurrent connectionist approach to normal and impaired routine sequential action. *Psychological Review*, *111* (2), 395-429.

Caruana, R. (1997). Multitask learning. *Machine Learning*, *28* , 41-75.

Chen, G., Saad, Z. S., Britton, J. C., Pine, D. S., & Cox, R. W. (2013, June). Linear mixed-effects modeling approach to FMRI group analysis. *NeuroImage*, *73* , 176–190.

Cooper, R., & Shallice, T. (2011). *The organization of mind*. Oxford, UK: Oxford University Press.

Cox, C. R., Seidenberg, M. S., & Rogers, T. T. (2014). Connecting functional brain imaging and parallel distributed processing. *Language, Cognition and Neuroscience*, 1âAS15. Retrieved from http://dx.doi.org/10.1080/23273798.2014.994010

Davis, T., & Poldrack, R. A. (2013). Measuring neural representations with fMRI: practices and pitfalls. *Ann N Y Acad Sci*, *1296* , 108-34.

Farah, M., & McClelland, J. L. (1991). A computational model of semantic memory impairment: Modality-specificity and emergent category-specificity. *Journal of Experimental Psychology: General*, *120* , 339-357.

Feredoes, E., Tononi, G., & Postle, B. (2007). The neural bases of the short-term storage of verbal information are anatomically variable across individuals. *Journal of Neuroscience*, *27* (41), 11003-11008.

Friston, K. J., Stephan, K. E., Lund, T. E., Morcom, A., & Kiebel, S. (2005, January).Mixed-effects and fMRI studies. *NeuroImage*, *24* (1), 244–252.

Harm, M. W., & Seidenberg, M. S. (2004). Computing the meanings of words in reading: Cooperative division of labor between visual and phonological processes. *Psychological Review*, *111* (3), 662–720.

Haxby, J. V., Gobbini, M. I., Furey, M. L., Ishai, A., Schouten, J. L., & Pietrini, P. (2001). Distributed and overlapping representations of faces and objects in ventral temporal cortex. *Science*, *293* (5539), 2425–2430.

Hinton, G. (1984). Distributed representations. *Computer Science Department*. Hinton, G. (2014). Where do features come from? *Cognitive science*, *38* (6), 1078–1101.

Hoerl, A. E., & Kennard, R. W. (2000). Ridge regression: Biased estimation for nonorthogonal problems. *Technometrics*, *42* (1), 80âAS86. Retrieved from http://dx.doi.org/10.1080/00401706.2000.10485983

Joanisse, M. F., & Seidenberg, M. S. (1999). Impairments in verb morphology after brain injury: A connectionist model. *Proceedings of the National Academy of Science, USA*, *96* , 7592-7597.

Kenward, M. G., & Roger, J. H. (1997). Small sample inference for fixed effects from restricted maximum likelihood. *Biometrics*, 983–997.

Kriegeskorte, N., Goebel, R., & Bandettini, P. A. (2006). Information-based functional brain mapping. *Proceedings of the National Academy of Sciences of the United States of America*, *103* (10), 3863–3868.

Kriegeskorte, N., & Kievit, R. A. (2013). Representational geometry: integrating cognition, computation, and the brain. *Trends in Cognitive Sciences*, *17* (8), 401–412.

Kriegeskorte, N., Mur, M., & Bandettini, P. A. (2008). Representational similarity analysis – connecting the branches of systems neuroscience. *Frontiers in Systems Neuroscience*, *2* . (PMID: 19104670 PMCID: PMC2605405)

McClelland, J. L., & Rumelhart, D. (Eds.). (1986). Cambridge, Massachusetts: MIT Press.

McClelland, J. L., & Rumelhart, D. E. (1985). Distributed memory and the representation of general and specific information. *Journal of Experimental Psychology: General*, *114* , 159-188.

Mitchell, T. M., Hutchinson, R., Niculescu, R. S., Pereira, F., Wang, X., Just, M. A., & Newman, S. (2004). Learning to decode cognitive states from brain images. *Machine Learning*, *57* (1), 145–175.

Mitchell, T. M., Shinkareva, S. V., Carlson, A., Chang, K.-M., Malave, V. L., Mason, R. A., & Just, M. A. (2008). Predicting human brain activity associated with the meanings of nouns. *Science*, *320* (5880), 1191–1195.

Norman, K., Polyn, S. M., Detre, G. J., & Haxby, J. V. (2006). Beyond mind-reading: Multi-voxel pattern analysis of fMRI data. *Trends in Cognitive Sciences*, *10* (9), 424-430.

Pereira, F., & Botvinick, M. M. (2011). Information mapping with pattern classifiers: A comparative study. *NeuroImage*, *56* (2), 476–496.

Pereira, F., Mitchell, T. M., & Botvinick, M. M. (2009). Machine learning classifiers and fMRI: a tutorial overview. *NeuroImage*, *45* (1, Supplement 1), S199–S209.

Plaut, D. C., McClelland, J. L., Seidenberg, M. S., & Patterson, K. (1996). Understanding normal and impaired word reading: computational principles in quasi-regular domains. *Psychological review*, *103* (1), 56–115.

Plunkett, K., & Juola, P. (1999). A connectionist model of English past tense and plural morphology. *Cognitive Science*, *23* (4), 463–490.

Rao, N., Cox, C. R., Nowak, R., & Rogers, T. T. (2013). Sparse overlapping sets lasso for multitask learning and its application to fMRI analysis. In C. Burges, L. Bottou, M. Welling, Z. Ghahramani, & K. Weinberger (Eds.), *Advances in neural information processing systems 26* (pp. 2202–2210). Curran Associates, Inc.

Rao, N., Nowak, R., Cox, C. R., & Rogers, T. T. (2014). Classification with Sparse Overlapping Groups. *CoRR, abs/1402.4512*.

Riggall, A. C., & Postle, B. R. (2012, September). The relationship between working memory storage and elevated activity as measured with functional magnetic resonance imaging. *The Journal of Neuroscience*, *32* (38), 12990–12998.

Rish, I., Cecchi, G. A., Heuton, K., Baliki, M. N., & Apkarian, A. V. (2012, February). Sparse regression analysis of task-relevant information distribution in the brain. In D. R. Haynor & S. Ourselin (Eds.), (p. 831412).

Rogers, T. T., Lambon Ralph, M. A., Garrard, P., Bozeat, S., McClelland, J. L., Hodges, J. R., & Patterson, K. (2004). Structure and deterioration of semantic memory: A neuropsychological and computational investigation. *Psychological Review*, *111* (1), 205–235.

Rogers, T. T., & McClelland, J. L. (2004). *Semantic cognition: a parallel distributed processing approach*. Cambridge: Mass. : MIT Press.

Rogers, T. T., & McClelland, J. L. (2014). Parallel distributed processing at 25: Further explorations in the microstructure of cognition. *Cognitive science*, *38* (6), 1024–1077.

Rohde, D. L. (1999). *LENS: The light, efficient network simulator*.

Rumelhart, D. E., & McClelland, J. L. (1986). On learning the past tenses of English verbs. In J. L. McClelland, D. E. Rumelhart, & the PDP Research Group (Eds.), *Parallel distributed processing: Explorations in the microstructure of cognition* (Vol. 2, p. 216-271). Cambridge, MA: MIT Press.

Rumelhart, D. E., & Todd, P. M. (1993). Learning and connectionist representations. In D. E. Meyer & S. Kornblum (Eds.), *Attention and performance XIV: Synergies in experimental psychology, artificial intelligence, and cognitive neuroscience* (p. 3-30). Cambridge, MA: MIT Press.

Seidenberg, M. S., & McClelland, J. L. (1989). A distributed, developmental model of word recognition and naming. *Psychological Review*, *96* (4), 523–568.

Smolensky, P. (1986). Information processing in dynamical systems: Foundations of harmony theory. In D. E. Rumelhart & J. L. McClelland (Eds.), *Parallel distributed processing: Explorations in the microstructure of cognition* (Vol. 1, p. 194-281). Cambridge, MA: MIT Press.

Tibshirani, R. (1996). Regression shrinkage and selection via the lasso. *Journal of the Royal Statistical Society. Series B (Methodological)*, *58* (1), pp. 267-288. Retrieved from http://www.jstor.org/stable/2346178

Zhou, J., Chen, J., & Ye, J. (2011). MALSAR: Multi-tAsk Learning via StructurAl Regularization [Computer software manual]. Retrieved from http://www.public.asu.edu/ jye02/Software/MALSAR

Zipser, D., & Andersen, R. A. (1988). A back propagation programmed network that simulates response properties of a subset of posterior parietal neurons. *Nature*, *331*, 679-684.

# Figure Captions

Figure 1: A PDP model used to understand semantic memory (from Rogers et al., 2004). Units in the *Visual* layer code visual features and units in the *Verbal* layer encode familiar words. The *Visual* and *Verbal* units can receive direct inputs from the environment, corresponding to direct perception of a visually-presented item or of a spoken statement. Units in both layers send connections to, and receive connections from, an intermediating hidden layer. To simulate a task such as object naming, visual features of the object are directly activated in the *Visual* layer and the activation propagates to other units via the weighted connections. If the weights are set to appropriate values, the model will ultimately activate the *Verbal* unit corresponding to the item name. Likewise name comprehension is simulated by directly activating the unit corresponding to the name and propagating activation throughout the network. With appropriate weights the visual features of the named item will activate, along with verbal units describing the item's properties. Appropriate weights are discovered through a predictive error-driven learning algorithm. Following learning, each input provokes a pattern of activation over hidden units that depends on the acquired weight configuration---a learned internal representation of the input. Though the particular pattern acquired for a given item varies across training runs, the representations always encode the same similarity structure among items in the environment, representing items that are conceptually related with similar patterns of activation.

Figure 2: Architecture of the auto-encoder network used to generate the data for 10 model subjects used in subsequent simulations. The model has 36 input units (18 systematic), 14 hidden units (7 systematic), and 36 output units (18 systematic). The 22 irrelevant units are completely disconnected from the network, and stand for units that subserve an unrelated function but are anatomically adjacent to units of interest.

Figure 3: Left: Ratio of between-domain to within-domain Euclidean distances for the representations coded over different sets of units in the network, averaged over the 10 model subjects. Distributed representations that encode the domain structure should have large distances for items from different domains and small distances for items from the same domain, and so should show a large ratio. While the systematic I/O units clearly code the domain structure to some degree, the systematic hidden units express the structure more strongly.

Figure 4: Results from the univariate analysis of simulated data. Bar height indicates the absolute value of the statistic for the unit-wise contrast between conditions at the group level. Colored bars indicate units showing significant differences with p-values corrected to control the false discovery rate at . The red-blue scale indicates the direction of the contrast effect across model subjects, with red indicating units consistently showing greater activation for items.

Figure 5: Result of the multivariate searchlight analysis of simulated data. Bar height indicates the mean classifier accuracy over subjects. Bars in red indicate searchlights where classification accuracy differed from chance over subjects, p-values corrected to control the false discovery rate at . Each row shows results from a different searchlight size indicated by the number on the far right. Mean and se indicate the mean and standard error of the classifier accuracy, respectively; n indicates the number of units whose searchlights shows significant information across model subjects.

Figure 6: The results from the Lasso analysis. The height of each bar indicates the number of times each unit was selected, that is, given a non-zero weight over subjects. Colored bars were selected more often than expected by chance given the overall rate of unit selection. The blueness or redness of the bar conveys the frequency with which each unit was assigned a positive weight over subjects. Positive weights mean that activation at that unit will push the model towards labeling the current item as belonging to domain A. Mean and se indicate the mean and standard error of the classifier accuracy, respectively; n indicates the number of units selected more often than chance across model subjects.

Figure 7: Hit rate and precision for different multivariate methods, computed across the whole network (top), the I/O units only (middle), and the hidden units only (bottom).

Figure 8: The results from the ridge regression analysis. Height of each bar indicates the mean across model subjects of the absolute value of the weight placed on each unit by the classifier. Bar hue indicates the frequency with which each unit received a positive weight across model subjects. Bar saturation indicates which units would count as “selected” under three different policies, based on weight magnitude. Bright bars are in the top third of the distribution, pale bars are in the middle third, and gray bars are in the bottom third. Mean and se indicate the mean and standard error of the classifier accuracy, respectively; n indicates the mean number of units with non-zero weights across model subjects.

Figure 9: A schematic laying out two hypothetical solutions, and the *set penalty* they incur. The set penalty is one of the components of the SOS Lasso regularizer. Note that in cases of overlap, the weight will be accounted for only once, as belonging to the set in which it increases the penalty by the smallest amount. In other words, SOS Lasso will consolidate weights into as few groups as possible.

Figure 10: The results from the SOS Lasso analysis. The height of each bar corresponds to the number of times each unit was selected over subjects. Colored bars were selected more times than would be expected given the overall rate of unit selection. The blueness or redness of the bar conveys the frequency with which each unit was assigned a positive weight over subjects. Positive weights mean that activation at that unit will push the model towards labeling the current item as belonging to domain A. Mean and se indicate the mean and standard error of the classifier accuracy, respectively; n indicates the number of units selected more often than chance across model subjects.

Figure 11: Results from applying different multivariate methods to the star-plus fMRI dataset. (A) The information maps yielded by applying different multivariate methods to the same “star-plus” dataset. Because the solutions are very sparse for Lasso and SOS Lasso, all voxels selected for any subject are plotted. For ridge regression, the plot shows voxels selected across all subjects under a lax and an aggressive elimination policy. For LASSO, ridge, and SOS Lasso, the blueness or redness of the dots conveys the proportion of the time each unit was assigned a positive weight over subjects. (B) Mean and standard error of the classifier cross-validation accuracy for each method across subjects in the star-plus dataset.

# Table Captions

Table 1: Assumptions implicitly adopted by different statistical methods for image analysis.

Table 2: A summary of the answers for each method across the four central questions. The success of the searchlight was contingent on model parameters that may be difficult to discern in practice. Ridge regression identifies all units as informative, including ones that in fact contain no information.